uc3m | Universidad Carlos III de Madrid

BACHELOR'S THESIS

Degree in Aerospace Engineering

Preliminary design of the autopilot of an autonomous flapping wing micro air vehicle

Mercedes García Pérez

June 2018

Supervisor

Óscar Flores Arias

Abstract

This Bachelor's Thesis studies the implementation of machine learning for the preliminary design of the autopilot of a MAV flapping wing vehicle. For this purpose, reinforcement learning is applied in this study through the development of a Q-learning algorithm. Due to the complexity that a flapping wing vehicle presents, the aim of this project is to design an autopilot that is able to reach longitudinal control of a vehicle whose aerodynamic model is almost unknown. In order to develop a tool that accounts for complete longitudinal control, intermediate autopilot designs have been carried out so that the introduction of degrees of freedom to the problem has been gradual. Throughout the project, there is special concern about the development of a time and cost effective tool.

KEY WORDS

Aerospace engineering, Aerospace control, Aerospace simulation, Autonomous vehicles, Unmanned aerial vehicles

Acknowledgements

In first place, I want to thank Óscar Flores, my supervisor, for his patience and dedication in his efforts to guide me throughout the development of the project. Thank you for providing me with so much support. Furthermore, I am grateful to all the professors that during these four years have worked so hard to transmit me and the rest of the students so much knowledge and wisdom.

In second place, but not less important, I strongly thank my parents for their support. They have given me love and all the opportunities they could, and I hope I have not let slip any. I want to encourage my brother to be my relay at university, becoming the next engineer in the family. Finally, I would like to thank Cristóbal for being the best discovery I have made at university.

Contents

List of Figures

List of Tables

1 Introduction

As the time goes by, it is getting more and more common to hear about drones, also called Unmanned Aerial Vehicles (UAV's). The main characteristic of this type of vehicles is that they do not carry any people inside, neither passengers nor crew. Originally, Unmanned Aerial Vehicles were developed for military purposes, more precisely, during the First World War [1]. The first UAV's were created by the US army in 1918 and were named "torpedoes", which were launched by catapult or remotely driven by radio control. Around 1935, the British Army developed UAV's in order to used them as targets for shot training. This is thought to be the reason why, colloquially, people also call UAV's as "Drones", since one of the UAV's model was called the DH82B Queen Bee. It was after the Vietnam War that this type of technology was further developed, looking for an extended endurance and greater altitude flights.

In the latest years, they have been widely introduced in the industry and because of their characteristics, their advantages and the improvement of technology, many companies and researchers are keen on developing drones that can make human life easier at work or for leisure. Their current applications in the industry include military as well as civil purposes. They can be used for the monitoring of an environmental mission or a natural disaster, for surveillance and reconnaissance purposes, for filming and even delivering. As one can imagine, the use of drones, mostly for military purposes, can become controversial.

With regard to the Legal Framework embracing drones, each country has its own regulations. The United States laws are established by the Federal Aviation Administration, while the use of drones in European countries is governed by the European Aviation Safety Agency, that states a case by case basis for this kind of regulations. In Spain, the last law that regulates UAV applications was approved in December 2017 [2]. According to this regulation, hospitals are now able to transport blood and urgent material avoiding traffic jams, security forces can use drones for surveillance and drones can now be flown in cities with an AESA authorization at night.

Focusing on the current evolution of drones, military forces are encouraging the development of smaller and more agile UAV's, which are easier to carry and can go unnoticed by the enemy's troops. This emergent type of drones are known as Micro Air Vehicles (MAV's). This technology has become achievable thanks to the further development of microprocessors and batteries. For instance, British troops are implementing the use of this kind of Unmanned Aerial Vehicles in missions in Afghanistan.

Drones can be classified according to different features. Regarding the size, one can find Micro Air Vehicles and at the same time one can distinguish between rotary, fixed or flappling wings as a propulsion system. The most common types of drones are quadcopters (rotary wing) and fixed wing UAV's. The dynamics and control of

a flapping-wing drone are still being developed since the aerodynamics of this type of vehicle are much more complex than for the case of a rotary wing drone, which is more stable, simple and predictable.

A clear example of flapping-wing is an insect, such as a fly. Think of all the maneuvers a fly can perform in a fraction of a second. If humans could control the motion of Micro-Air Vehicles using flapping wings, many more applications could be attributed to drones, for instance, in the field of biomimetics. Flapping wing Micro Air Vehicles would be the next step regarding UAV development [3]. It is fascinating that this ancient and natural propulsion system is still nowadays becoming a challenge for humans to understand and control.

1.1 Definition of the problem

When birds or insects flap their wings, they generate aerodynamic forces that let them perform the desired maneuvers. The thrust originated by the wings has to counteract the forces of drag and weight. There are many parameters that account for the thrust that is going to be generated and that affect the motion of a flapping wing vehicle. Moreover, it is actually complex to define and control those parameters and study their impact when flight dynamics are non linear, which is the case of a flapping wing vehicle.

The control of this type of vehicle becomes a challenge for engineers and designers. There is much information about the control of a rotary wing drone, but little is found in literature about the control of a flapping wing one, for which complexity comes into play.

A possible way to deal with this complexity is the creation of an autopilot for which the aerodynamic model of the flying object is not completely defined. This autopilot could be applied to the case of a flapping wing vehicle, since it would be able to control the vehicle with a very reduced knowledge about its aerodynamics. In this way, a flapping wing vehicle could be capable of performing any maneuver without anybody establishing a sequence of the control parameters, being automatically controlled.

1.2 Objectives

The main objective of this project is to create an autopilot that is able to control a Micro Air Vehicle whose aerodynamic model is not completely understood or defined, which would be the case of a flapping wing MAV. For this purpose, the control parameters that define the motion of a flapping wing vehicle will be autonomously driven with the support of a machine learning method using a MATLAB@ algorithm. This project adapts the Q-learning algorithm to create a new one in which a flapping wing MAV can autonomously learn how to reach longitudinal control with the

variation of thrust and the stroke plane angle as a first approach. The proposed autopilot will be a preliminary design.

Once this tool is developed, the control parameters can be defined for the design of an MAV and their effect on the motion can be known and controlled. Furthermore, the vehicle will be able to perform any maneuver autonomously just introducing the desired trajectory to follow. In a real situation, an MAV could be trained in this way if the tool is further developed to the six degrees of freedom of a real rigid body.

Moreover, the second objective of this project to minimize the costs of the production of this preliminary autopilot design in order to save time and efforts.

1.3 State of the art

Regarding a nature example of flapping wings, flying insects rely on the provision of sensory feedback. The inputs of sensors such as compound eyes, ocelli and antennae are used as feedback to the insect flight control system. When it comes to drones or flying machines, a similar method is applied, but in this case the visual information or input is obtained from accelerometers, gyroscopes and sensors. This information is translated into functional forms for state feedback. Inertial sensors play a key role when it comes to stabilization.

Thinking about nature and evolution, how did insects and birds learn how to fly? Interaction with the environment is the first thing that comes to mind [4]. For instance, eagle baby birds, or eaglets, firstly observe their parents flying close to the nest. Then, they start jumping from branch to branch near the nest flapping their wings, exercising their ability to maintain equilibrium and coordination. Once eaglets are strong enough, they fly from branch to branch or even to other trees until they are finally able to fly thanks to all the training. This training is based on a trial and error basis, which results in a progressive learning.

Machine learning is a way of learning based on this principle. Actually, the University of Bristol in collaboration with BMT Defence has already performed the first perched landing of an UAV using machine learning algorithms [5]. Inspired by bird wings, the structure of a fixed wing was converted into a moving wing, through which they wanted to extend the operation of this type of vehicle. In this way, they would create an UAV's that would be more efficient and helpful, for instance, providing aid in a humanitarian disaster. After some modifications on a fixed wing, the flight dynamics of the UAV became non-linear. In order to deal with this new complex dynamic model, they made a choice on using machine learning. They collected data from the performance of the UAV in wind tunnel experiments and, using it as a database, the UAV was able to know how to behave in order to achieve a perch landing performance. They support that machine learning allows the possibility of controlling highly manoeuvrable UAVs for non linear dynamics.

Regarding the flight control of current aircraft and UAVs autopilots, the most

common solution is the use of P, PD or PID controllers. These types of controllers can be applied for kinematic control of an UAV as it is explained in [6]. It is very common to find this type of controllers for fixed wing UAV's. The preferred type of controller is the Proportional-Integral-Derivative (PID), which is also the most expensive. Despite being a linear controller, it still works for a non linear system. The principle of this controller is to minimize the error between the desired trajectory to follow and the current trajectory that the UAV is describing through the application of a Closed-Loop System. Another suitable UAV control method is explained in [7], the Sliding Mode Control, which is a non linear controller that has its origin in its application for marine vehicle control.

The problem with PID controllers and Sliding Mode Control for its application in flapping wing vehicles is that they require previous knowledge about the behaviour of the system. As it has been said, there is little information about the dynamic model of a flapping wing vehicle. Therefore, these are not valid solutions for the proposed problem of this project.

1.4 Socioeconomic impact

As it has been said, Unmanned Aerial Vehicles have already been introduced into human life at work and at leisure time. The range of applications that involve this type of vehicles is very wide: military and security activities such as surveillance and reconnaissance, artistic purposes such as movie filming and photography, transport of light objects,... As it can be noticed, all these applications are very useful and are in most of the cases, beneficial for human life if a proper use is made of UAV's.

Unmanned Aerial Vehicles seem very attractive for delivering companies since the incorporation of this type of vehicles would reduce the time and costs spent on displacement. As a consequence, the implementation of these vehicles would reduce the need of human workforce and air traffic would get uncontrolled, leading to a threatened public safety. The implementation of new technologies also make people assume some drawbacks.

With regard to the application of flapping wing vehicles, bio-mimetic robots could have applications related to espionage and nature filming due to their small size, great manoeuvrability and physical similitude with insects.

2 Methodology

2.1 Q-Learning

Reinforcement Learning is a type of machine learning, based on the Markov Decision Process. When a dog is being trained how to shake anyone's hand, for instance, it collects cause and effect information in a trial and error process: the dog is given a snack when it succeeds and a penalty when it does not. That is the principle of reinforcement learning. The elementary solution methods are Dynamic Programming, Monte Carlo Methods and Temporal-Difference Learning (TD). The last one includes algorithms such as Sarsa, Q-Learning, Actor-Critic Methods and R-Learning (For further information check [8]).

In fact, using machine learning lets the MAV learn by interaction with the environment in a trial and error learning process. The purpose of the problem is to find a solution for which no information about the dynamics of the MAV needs to be previously known for the design of its autopilot. The type of reinforcement learning that better fits the characteristics of the problem is TD Learning as it does not need a model and it can perform an incremental computation. Dynamic Programming requires an accurate model of the environment and Monte Carlo Methods do not require a model but it cannot perform a step-by-step integration, so that these methods are not valid for the description of the proposed problem.

Temporal-Difference Learning is based on the development of a policy. A policy is defined as a definite course or action adopted for the sake of expediency. Regarding TD Learning solutions, these methods are classified in On-policy (Sarsa and Actor-Critic Methods) and Off-Policy (Q-Learning and R-Learning). On-policy methods predict the return for the current policy whereas Off-policy methods drive the policy to improve locally with respect to the current policy. In other words, On-Policy methods evaluate the policy that has been used to make decisions whereas Off-Policy methods evaluate or improve a policy that is different from the one that was used to generate the data. An Off-Policy algorithm will better suit the purpose of the desired solution since there is not a starting policy map to be improved. The policy map in this project is built from scratch.

The main difference between Q-Learning and R-Learning is that Q-Learning optimizes discounted reward and R-Learning optimizes average reward. An optimization of discounted reward makes far-future rewards less important than short-term reward whereas optimizing average reward equally weights future and short-term rewards. In Section 3 it will be shown that discounted reward is needed for a more effective algorithm. Moreover, R-Learning has been less explored than Q-Learning and it should be considered experimental. Therefore, a decision has been made about using Q-Learning for the design of the autopilot of the flapping wing MAV since the analysis of the algorithm is simple and it allows a rapid convergence.

Q-Learning is based on the construction of a policy map in the form of a matrix Q. This matrix has as many rows as states (N_s) and as many columns as the number of possible actions (N_a) that are defined, so that $Q_{N_s X N_a}$. These parameters depend on the complexity of the problem and the memory of the used CPU since a too large Q matrix could exceed the amount of available RAM memory. The aim of the policy matrix is to know by looking at element $Q(i, j)$ how good action j is when the vehicle is at state i.

2.2 Application of Q-Learning

In order to tackle the proposed problem, the development of a policy map will experience two main phases that will take place in isolated algorithms: a first phase of exploration and a second phase of exploitation. The aim of the exploration phase is, as its names indicates, to 'explore' and, consequently, to build from scratch the policy map. After the exploration phase has concluded, the exploitation phase tests the resulting policy map through the performance of a specific mission just by the execution of the policy map.

2.2.1 Exploration algorithm

As it has been previously said, the aim of this algorithm is to build the policy map for the proposed problem. This algorithm converts a matrix of zeros into a policy map through an iterative updating process. In other words, this is the training phase.

In order to calculate the values of the policy map, a reward function R has to be defined, so that the algorithm knows what does 'good' mean. In the case of this problem, the value of R has to do with how far the current state of the vehicle is from the training state. Therefore, it can be said that the reward function is a function of the state s (Equation 1). It will be presented as minus the quadratic error from the target trajectory, so that it is desirable to get the maximum reward possible (negative number of small absolute value).

$$R = f(s) \tag{1}$$

The steps of the algorithm are the following. Once the state of the vehicle is known, it is the time to take an action. Should the vehicle explore or should it do what the policy map establishes to be the best action? The exploration algorithm has been designed in a way that there is 50 % probability of choosing a random action and 50 % probability of choosing an action according to what the policy map has found to be the best option at the moment. In any case, the executed action will always belong to the available set of actions established at the beginning of the algorithm.

Once the action has been chosen, the dynamics of the problem come into play. The equations of motion are integrated and so the new state s_{t+1} is obtained from the old state s_t. With this information, the Q matrix is updated. This update of the Q matrix is influenced by several parameters that are characteristic of the Q-learning algorithm. In order to identify them, one should know that the update of matrix Q follows the expression

$$Q(s_t, a_t) = Q(s_t, a_t) + \mu[R(t+1) + \gamma max(Q(s_{t+1})) - Q(s_t, a_t)] \tag{2}$$

Eq. 2 shows the base of the Q-Learning algorithm. The update of matrix Q depends on both the current s_t and the future state s_{t+1} as well as on the learn rate μ and the discount rate γ. By definition, the learn rate represents how is the new value estimate weighted against the old one ($\forall \mu \in [0, 1]$), meaning that for $\mu = 1$ all new values are taken into account and that should be acceptable for no noise situation. Furthermore, the discount rate represents how important is the value of the future state ($\forall \gamma \in [0, 1]$). For the moment, there is actually no previously established value for the learn and discount rates that assures that the algorithm performance is the best one. Therefore, some values will be estimated and assigned in Section 3 according to the parametric study that will be carried out in that section.

The development of a policy maps follows an iterative process. In the exploration algorithm, the policy map is updated every time an action is executed in order to reach a progressive learning process. This phase stops when the Q matrix is not being updated any more.

2.2.2 Exploitation algorithm

In the exploration or training phase, the vehicle is taking random actions and actions according to the policy map indiscriminately. Therefore, it is very difficult to measure the quality of the policy map since intuition is missed. That is the reason behind the creation of the exploitation algorithm.

The goal of the exploitation phase is to test the policy map that has been obtained in the exploration process and have a quantitative measure of the quality of the policy map. In this second phase, the policy map is not being updated. All actions will be taken according to the policy map. In this way, the real performance of the algorithm will be visible, making it easier to evaluate how 'good' is the tested policy map.

2.3 Time integration of dynamics

Since the analytical solution of the equations of motion can become a very difficult task, they will be solved with the application of a numerical method. In this project, the numerical method that will be used is the Fourth Order Runge Kutta method, which is one of the most widely used. This type of methods are applied to an initial value problem. In the proposed problem, $\vec{F}$ corresponds to the equations of motion, being a function of the state vector $\vec{q}$ and time t

$$\frac{d\vec{q}}{dt} = \vec{F}(\vec{q}, t) \tag{3}$$

for an initial condition

$$\vec{q}(t = 0) = \vec{q_0} \tag{4}$$

The RK4 method solves the problem through an explicit iterative process

$$\vec{q_{i+1}} = \vec{q_i} + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \tag{5}$$

where h is the time differential, defined as $h = t_{i+1} - t_i$ and tke k_m values are defined as the following slopes

$$\left.\begin{aligned} k_1 &= \vec{F}(t_i, \vec{q_i}) \\ k_2 &= \vec{F}(t_i + \tfrac{1}{2}h, \vec{q_i} + \tfrac{1}{2}k_1 h) \\ k_3 &= \vec{F}(t_i + \tfrac{1}{2}h, \vec{q_i} + \tfrac{1}{2}k_2 h) \\ k_4 &= \vec{F}(t_i + \tfrac{1}{2}h, \vec{q_i} + k_3 h) \end{aligned}\right\} \tag{6}$$

According to this method, the duration of a time step h has to be established before the simulation making sure that the numerical solution obtained through this integration process is close to the analytical one.

3 One Degree of Freedom Problem

As it was said in the introduction, the dynamics of a flapping wing MAV are going to be studied in this Bachelor's Thesis. In order to understand the performance of the Q-Learning process in a simple example, this section will study the case for which the MAV only has one degree of freedom (d.o.f.): along the vertical axis z. This one d.o.f. problem will be considered as a 'toy' problem that will be useful in order to understand the concept of a policy map as well as the roles of both the exploration and exploitation algorithms and to set the Q-learning parameters (μ, γ) that optimize the construction of the policy map with the support of a simple example.

3.1 Definition of the problem

There are three reference frames involved in this problem: an inertial reference frame at a point I, a reference frame located at the target position O that moves with respect to the inertial one and a reference frame which has its origin at the center of gravity G of the MAV. For all the reference frames the z axis is positive pointing upwards.

It is known that when it comes to a drone, the thrust cannot be directly controlled, but the parameters that affect it, which are the control parameters. For simplicity, in this case there are three possible actions, which just involve maximum thrust up, maximum thrust down and zero thrust. The complete dynamic model is found in Figure 1.

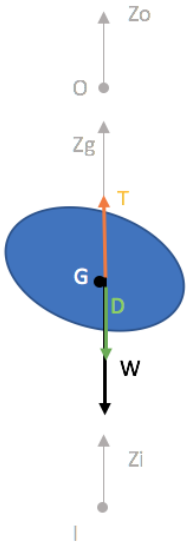


Figure 1: Dynamic Model for a one dimensional movement

Following Newton's Law in the z axis (Eq.7)

$$m\frac{d\dot{z}}{dt} = \sum F_z \tag{7}$$

it is possible to derive the equation of motion.

As it has been said, this problem will be considered as a toy example problem. Therefore, for the sake of simplicity, it will be treated as a non-dimensional problem. The non-dimensional equation of motion of this toy problem (Eq. 8) implements the contribution of a non-dimensional thrust force and a non-dimensional friction friction force that is linearly dependent on the speed of the vehicle

$$\ddot{\tilde{z}} = -\tilde{z} + \tilde{T} - C_f\dot{\tilde{z}} \tag{8}$$

Once the dynamic model has been established, it is the moment to introduce it into the algorithm. As it was previously said, two different algorithms are used in this problem: the first one for exploration and the second one for exploitation. In the exploration algorithm the target and the inertial reference frames are coincident. The MAV is, therefore, trained to reach the zero speed and position with respect to the inertial reference frame starting from any state included in the discretization range.

Remember that the reward function has to be established in order for the algorithm to evaluate how 'good' is the position of the MAV with respect to the target trajectory. In this case the reward function has been defined as the quadratic error from the target position and speed (Equation 9). As it has been said, for the exploration algorithm the target position and speed are $z_0(t) = 0$ and $\dot{z_0}(t) = 0$, respectively.

$$R(t) = -\phi|\tilde{z}(t) - \tilde{z}_0(t)|^2 - (1-\phi)|\dot{\tilde{z}}(t) - \dot{\tilde{z}}_0(t)|^2 \tag{9}$$

R is always negative and the smaller absolute value the closer the MAV is from the target position. The ϕ parameter is used here to define whether there is more concern about reaching the desired position penalizing the final speed or vice-versa. The case of $\phi = 0.5$ will be the case for which there is balance. The optimum value for ϕ will also be assessed in this section.

As it was said above, the aim of this section is to study the influence of the values of the learn and discount rates and assign the optimal ones to the following cases of study. But first, something has to be clear, how can one know when the algorithm has stopped learning? It would not be efficient to run the algorithm for longer than it needs and the results may not be accurate if the number of episodes is insufficient. The way to avoid this will be later explained.

3.2 Definition of the vehicle and conditions

Before the learning and exploitation algorithms are applied, the vehicle characteristics and the environment conditions have to be established. The one d.o.f. case is a first approach model used for the definition of the learning algorithm parameters. Therefore, and for simplicity, the friction coefficient has been set to be the following

- $C_f = 0.1$

The set of available actions for the non-dimensional thrust is maximum thrust up, maximum thrust down and zero thrust, as it has been previously said. The maximum value of thrust will be set at $T_{max} = 2$, a value that does not limit the performance of the vehicle. This set of actions is complete enough to allow all desired maneuvers in one dimension: hover, upward acceleration and downward acceleration.

Moreover, the learning state space of the MAV is compound within the following ranges. Remember that this is a non-dimensional problem, so that position and speed have no dimensions here

- $-2 \leq \tilde{z} \leq 2$
- $-2 \leq \dot{\tilde{z}} \leq 2$

It may be considered that the discretization of the state space of the MAV is mainly dependent on its application. It can be thought that if more accuracy is needed, the smaller position range should be explored and the greater the resolution applied (number of elements in the discretization). The effect of resolution and the range of the state variables will be explicitly studied in later sections. It will be crucial for the further development of the autopilot.

As a first approach and for redundancy, a margin of two above and below the target position has been chosen since the position range in the exploitation algorithm will be set to be smaller. The same rule is applicable to speed. According to this criteria, if the MAV was forced to move in a shorter range, the range in the state space could also be reduced. Regarding the range for the speed, the target speed will be no longer than one in the exploitation algorithm. Therefore, the chosen range is more than enough to keep the MAV at a small relative speed with regard to the target one. However, in order to check this, a specific study about the influence of the range of the state variables will be carried out in Section 4.9.

A resolution of 103 elements in the position range and 63 elements in the speed range has been applied. This resolution is going to produce a very accurate and rich policy map. It is affordable since the size of the Q matrix does not represent a challenge for the CPU memory; it is a matrix of 6489 rows and 3 columns. A further study of resolution will be carried out in Section 3.6.

In order to make the learning process more time efficient, a learning episode is automatically finished if the MAV has reached a zone that is considered as success

or a satisfying state. In this particular case, the imposed condition is

$$\sqrt{\tilde{z}^2 + \dot{\tilde{z}}^2} < 0.03 \tag{10}$$

This condition has been applied to the mentioned resolution, case for which the learning episode will finish if the norm of the new state is smaller than 0.03, which is an acceptable error compared to the size of the state space. This policy makes the zone inside the threshold be unexplored, which can become a disadvantage for rougher resolutions.

As a first approximation, it has been considered that the vehicle is capable of changing of action with a non-dimensional frequency of 10 since this type of vehicle may show a long time of reaction. Moreover, regarding the integration of the dynamics of the problem, the time step has been set to be $h = 0.05[-]$ in the RK4 method explained in Section 2.3. The justification for this choice can be found in Appendix A.

3.3 Effect of the learn rate

The first parameter that is going to be analyzed is the learn rate, since it seems to be the simplest parameter or, at least, the most intuitive. Maintaining unchanged the values for ϕ and the discount rate, different values for the learning rate will be studied: $\mu = 0.1$, $\mu = 0.5$ and $\mu = 0.9$. The case of unity will not be studied since some noise is expected. The analysis will be carried out for fixed values $\phi = 0.5$ and $\gamma = 0.9$.

Firstly, the exploration phase is carried out in order to build the policy map. As it was said, before stopping the exploration process of the policy map, one needs to be sure that the algorithm has finished learning. For that purpose, the evolution of the values of the Q matrix in four zones of the state space during the learning process has been shown in Figure 2.

In order to study the different zones, the value of Q associated to each action (total thrust up, total thrust down or zero thrust) in each zone will be the mean of the respective values of all the states that compose the zone. Since the number of available actions is three, there are going to be three different curves for each zone, one per action. Among the curves that correspond the same zone, the one that gets horizontal at the minimum absolute value represents the action that the policy map considers the best choice when the vehicle is found at that zone. Once the episodes keep running and the mean values for the update of each zone in Q do not change (when the lines get horizontal), the Q matrix is not being updated anymore. Table 1 shows the color criteria for the representation of each of these zones of study. Moreover, Figure 2d will be useful in order to recognize each zone faster during this and the following sections.

Green	Red	Blue	Black
$\tilde{z} < 0$	$\tilde{z} < 0$	$\tilde{z} > 0$	$\tilde{z} > 0$
$\dot{\tilde{z}} < 0$	$\dot{\tilde{z}} > 0$	$\dot{\tilde{z}} < 0$	$\dot{\tilde{z}} > 0$

Table 1: Color criteria for the four zones studied for convergence

Looking at Figure 2, it is concluded that each zone is converged when their respective color lines get horizontal. There are remarkable similarities between the different values of learn rate. For all cases, the green and blue zones are the ones that converge the first. Therefore, it seems that the first thing that the MAV learns is what to do in order not to fall down. Furthermore, it can be appreciated that the lower the learn rate, the smoother the lines. The absence of oscillation in the evolution of the mean values of the Q matrix may lead to the conclusion that the algorithm is 'convinced' that what it is learning is 'good'.

Moreover, when the mean values of the Q matrix get steady, it can be appreciated that the red lines are always above the blue ones, the blue ones above the black ones and the green ones at the bottom. The absolute value of each zone has to do with how easy it is for the vehicle to reach the desired state when it is found in that zone. In this case, the vehicle is being trained to reach equilibrium at zero position and speed.

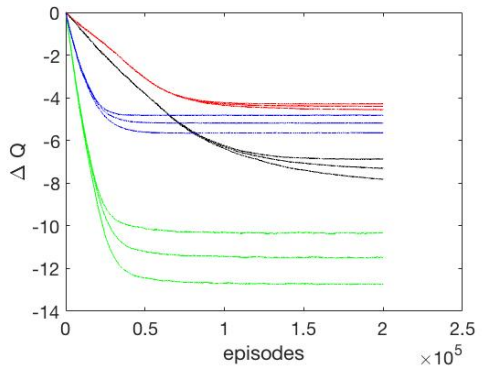
In this way, the worst scenario the vehicle can find is to be in the green zone since, naturally, the vehicle trend for zero thrust is to fall down according to Equation 8, moving away from the desired position. Therefore, this zone should show the greatest penalty.

There is another zone at which the vehicle is getting away from the zero speed and position state: the black zone. However, the penalty of being in this zone is lighter than in the green one. Why? As it has been said before, the natural trend of the vehicle is to fall down, which would decelerate the vehicle if it is at a positive position with positive speed, helping it to reach the desired state.

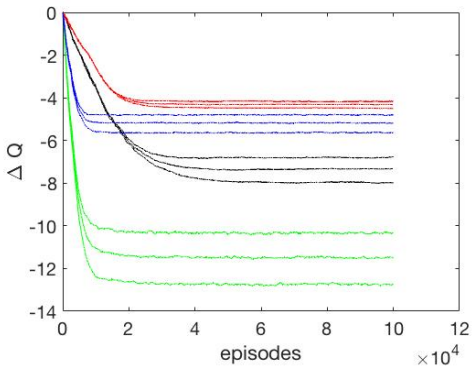
In the red and blue zones the vehicle is approaching the zero position, so the penalty is smaller than in the previous scenarios, since smaller corrections would be needed. Nonetheless, since the natural trend of the vehicle is to fall down, if the vehicle is found at the blue zone, it would pass over the desired position, reaching the green zone. If the vehicle is in the red zone, having positive velocity, it would decelerate while it is moving upwards, thus being it easier to reach zero speed and zero position. It is naturally easier for the vehicle to reach the desired state when it is in the red zone rather than in the blue zone. Therefore, the absolute value of the red zone should be smaller than the blue one, since the penalty is lighter and less corrections need to be made.

Another point in common between the three learn rate cases is that the first zones to converge are the ones of negative speed. This means that the first thing that the

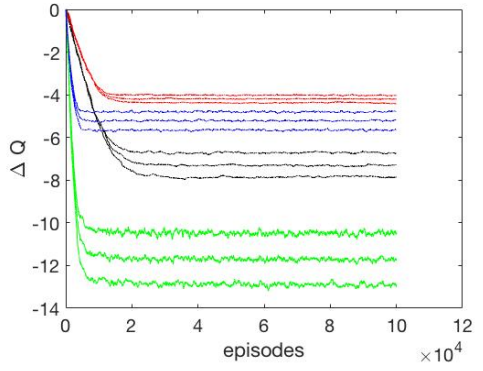
vehicle learns is not to fall down, which would in any case move the vehicle away from the training state.



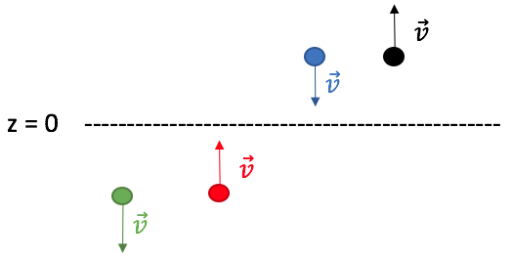
(a) $\mu = 0.1$



(b) $\mu = 0.5$



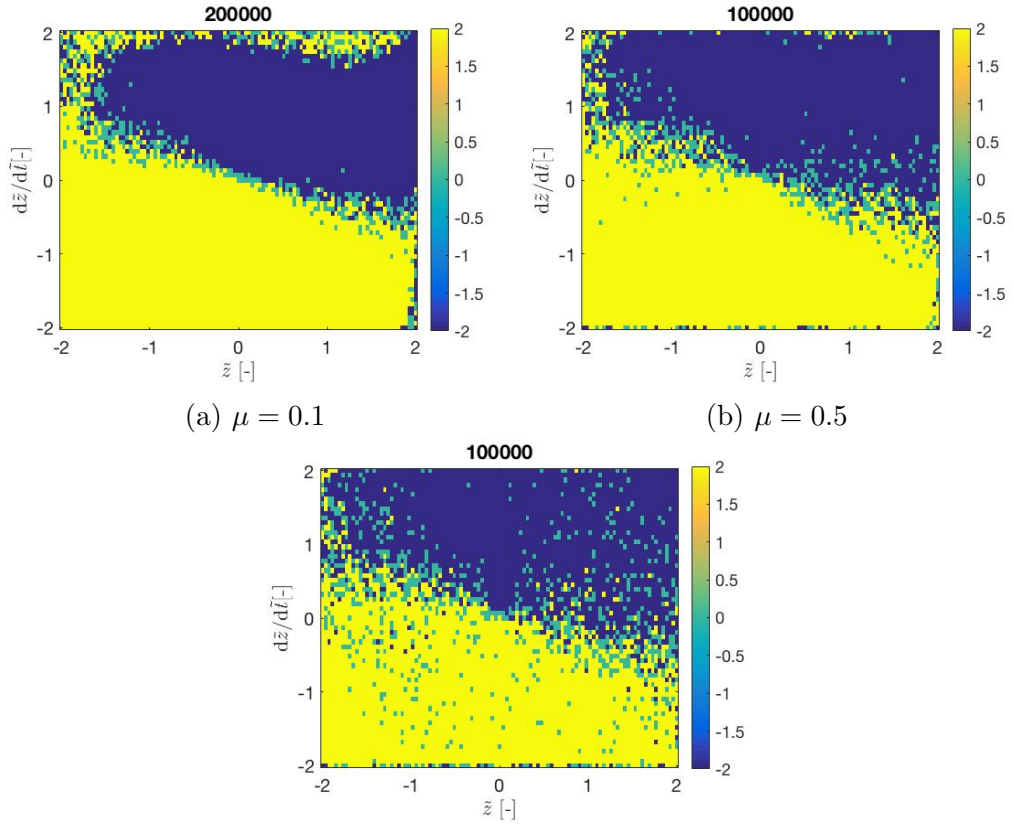
(c) $\mu = 0.9$



(d) Sketch of the four zones studied for convergence

Figure 2: Evolution of the values of the Q matrix for changing learning rate

Figure 2 shows that, as it should be expected, the larger the learn rate, the faster the convergence of the values of the Q matrix. However, a faster convergence does not necessarily mean that the resulting policy map is the one that works better. Therefore, in order to check the quality of the policy map, the exploitation algorithm will be later used. The number of episodes that have been performed during the exploration phase has been 200000 for a learning rate of $\mu = 0.1$ and 100000 for $\mu = 0.5$ and $\mu = 0.9$ since for those numbers the exploitation algorithm shows convergence. It can be appreciated that for a learn rate of 0.1 the black zone has almost converged. Furthermore, the black zone is the last one to converge in all cases. Having positive position and speed seems to be the most difficult state to make a choice about since the vehicle is reaching zero speed while it is moving away from the desired position.



(a) $\mu = 0.1$

(b) $\mu = 0.5$

(c) $\mu = 0.9$

Figure 3: Policy map representations for changing learn rate

As it has been defined before, a policy map has the form of a matrix, which is just a cell structure full of numbers. Therefore, if one wants to compare the different policy maps, plotting the elements of the Q matrix will not give any result that can be easily commented. Alternatively, the policy map will be represented in a more intuitive way. The best action to take is a function of speed and position. Therefore, a colour map has been used in order to represent the policy map. For each state included in the state space, the best action to take from the available set will be represented with a colour dot. For number i state, the best action to take is the one that gives the maximum number in the ith row of the Q matrix. A yellow dot indicates that the best action to take is maximum thrust up, a blue dot indicates maximum thrust down and a green dot indicates zero thrust.

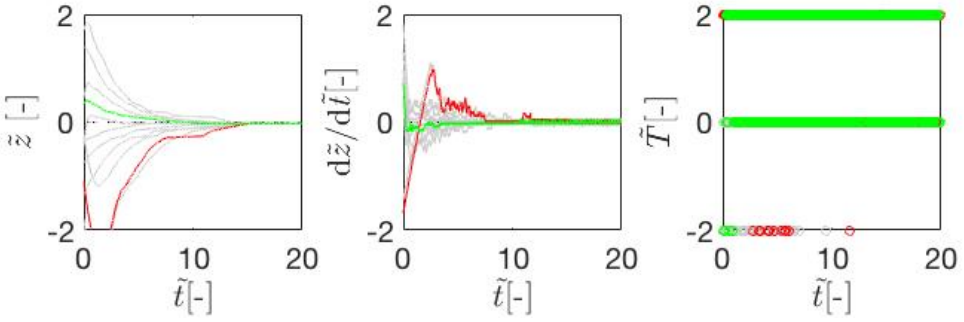
The obtained policy maps are represented in this way in Figure 3. There are similarities between them since they all show that if the MAV is below the target velocity, the best action to make is to use maximum thrust up (yellow) and if it is above, it is more appropriate to use negative (blue) or zero (green) thrust. That would be the logic behaviour to follow, so that the learning process shows consistency. Moreover, all the policy maps show a kind of inclined line between the yellow and the blue zones. In fact, if the vehicle is at a very large positive position with a

very low negative speed, it may be desirable to take maximum thrust down in order to accelerate the MAV towards the zero position. One may think that it could be enough for an actual drone just to fall, but regarding the range of actions of the vehicle it is more effective to use thrust down. The opposite case occurs when the vehicle is at a very negative position with low positive speed.

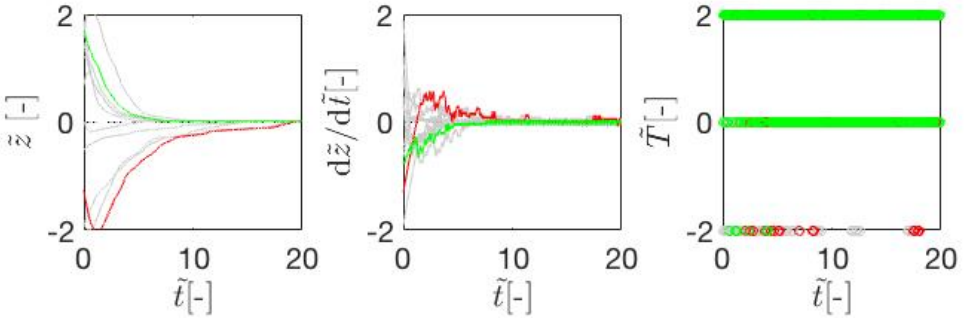
Apart from the similarities, the differences between the policy maps in Figure 3 are relevant. It seems that the policy map for a learn rate of $\mu = 0.1$ makes more sense for the case of maximum negative speed. For that case, one would expect the use of maximum thrust up in order to reach zero or positive speed. However, for learn rates of 0.5 and 0.9 the action to take at that situation is not clearly defined. Furthermore, it can be appreciated that the greater the learn rate, the more blurred the policy map. Maybe this issue could be relevant in order to check the quality of the policy map, but it will be better analyzed applying the exploitation algorithm.

Now that the learning process has finished, it is the time to run the exploitation algorithm and quantitatively measure the performance of the policy maps that have been obtained from the exploration phase. The vehicle has been trained to reach the zero position with zero speed, so Figure 4 shows the performance of each of the policy maps for the three different learn rate values for a target condition equal to the training condition in the exploration algorithm: $\tilde{z} = 0, \dot{\tilde{z}} = 0$. The criteria used to measure the quality of the performance will be the Mean Squared Error (MSE) between the target trajectory and the described one. The curves in green represent the sample episode with the smallest Mean Squared Error in position, whether the sample episode with the largest MSE is plotted in red. It can be appreciated that the algorithm has succeeded in order to train the vehicle reaching that state in all cases.

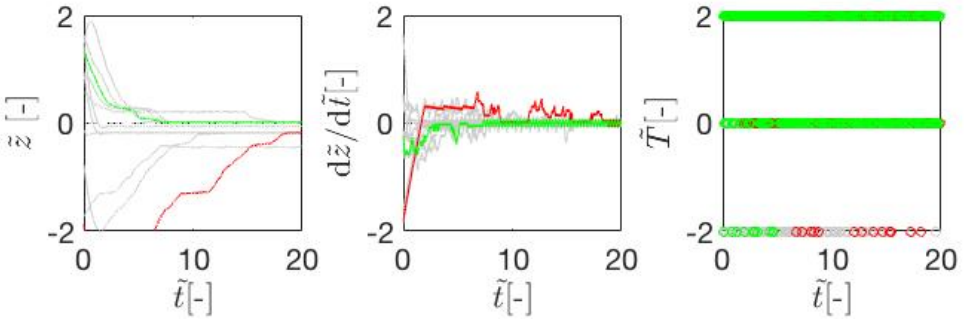
The performance test can be taken a step further. The vehicle has been initially trained to go from a random point in the state space to the zero position and speed point. Nonetheless, the purpose of an autopilot may not be just reaching a fixed point in the space, but following a moving target or a predefined trajectory. Then, the exploitation algorithm has been applied in a way that the origin of coordinates of the trained state space of the MAV is located at the moving target in order to make it the point to reach. In this case, at an instant t the reward is equal to the quadratic error between the current state of the MAV and the target one at that instant. The MAV also starts in this case at a random point inside the state space defined in the algorithm. The first eight seconds of the performance have been ignored for the calculations of the position and speed errors since it has been considered as the setting time. It will be considered that having a mean position error of 10 after stabilization means a crash of the vehicle, a mean position error smaller than one means not a crash but it is still a poor performance and a mean error smaller than 0.3 is a satisfactory performance.



(a) $\mu = 0.1$



(b) $\mu = 0.5$



(c) $\mu = 0.9$

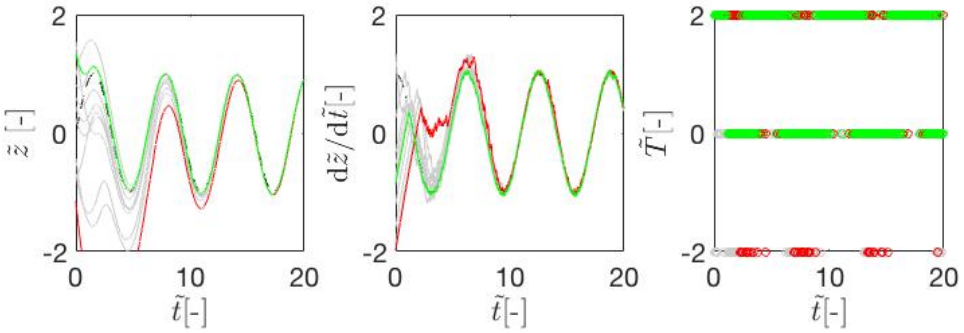
Figure 4: Performance of the vehicle trying to reach the training condition for changing learn rate. In green, the episode with the smallest position error; in red, the episode with the largest position error

Figure 5 shows how the MAV tries to reach a moving target that has been defined as a simple harmonic motion of one meter amplitude and one radian per second angular frequency. Looking at 5b it seems that the position error is minimized for $\mu = 0.5$. However, table 2 shows that the mean position error after the stabilization period is minimized for a learn rate of $\mu = 0.1$, case for which it is never larger than 0.3. Actually, for that value of the learn rate the mean position error never exceeds 0.1. As it can be observed for this application, the position and speed ranges that have been established for the discretization are more than enough since the range of the target trajectory is smaller than the range of the state space (Figure 5). The vehicle is always inside the state space, which means there is no moment at which

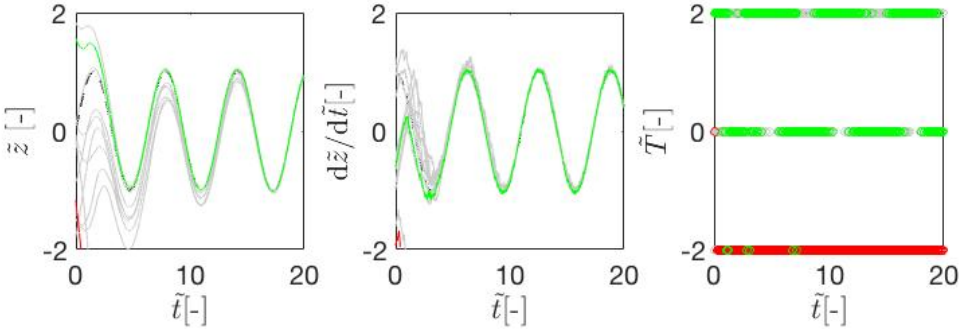
the policy map is not being efficiently used.

Learn rate	Error< 0.3	Error< 1	Error< 10
0.1	100	100	100
0.5	78.13	79.31	86.79
0.9	72.16	77.10	77.77

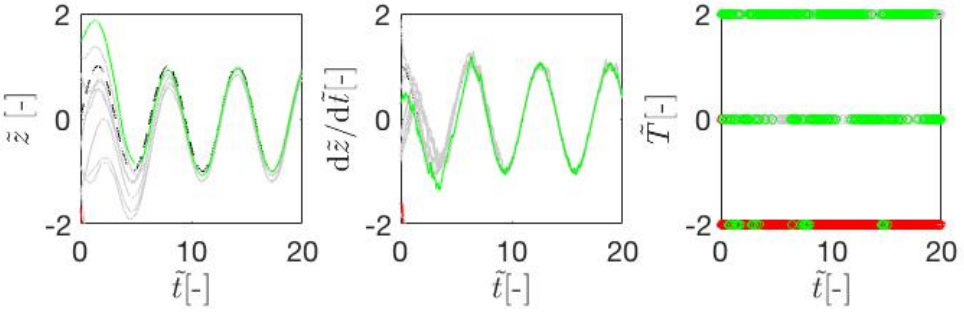
Table 2: Percentages of the number of episodes for which the mean position error after settling time is smaller than 0.3, 1 and 10 for changing learn rate. Simple harmonic motion. Total number of episodes: 10000



(a) $\mu = 0.1$



(b) $\mu = 0.5$

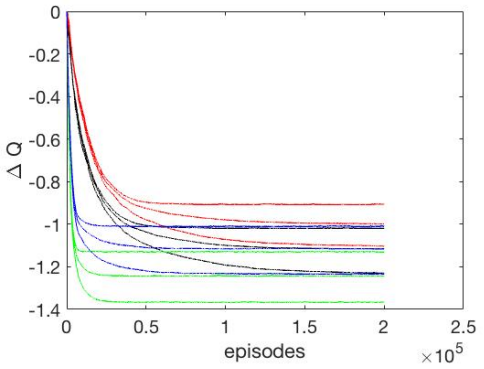


(c) $\mu = 0.9$

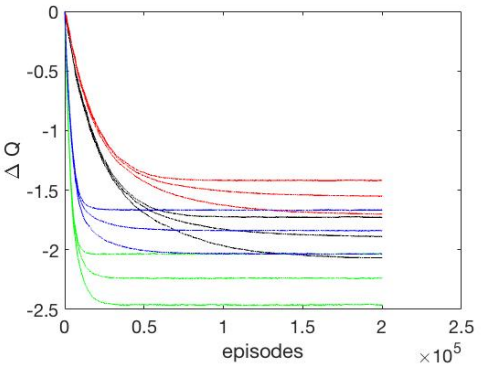
Figure 5: Evolution of position, speed and actions for changing learn rate. Simple harmonic motion. In green, the episode with the smallest position error; in red, the episode with the largest position error

3.4 Effect of the discount rate

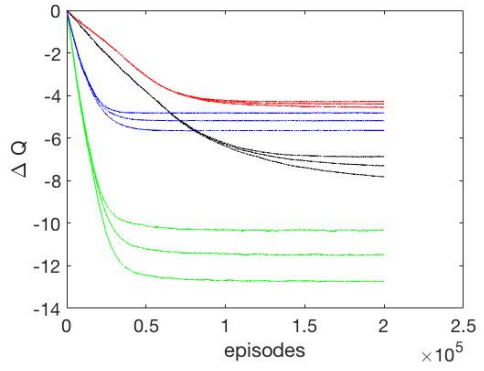
The following parameter to study will be the discount rate, since it is the other parameter that is explicitly involved in the Q learning algorithm. With regard to the discount rate, the optimal value for the learning rate is applied so that $\mu = 0.1$ and $\phi = 0.5$ have been fixed. Remember that the discount rate represents how important is the Q matrix value of the future states in the update of the policy map.



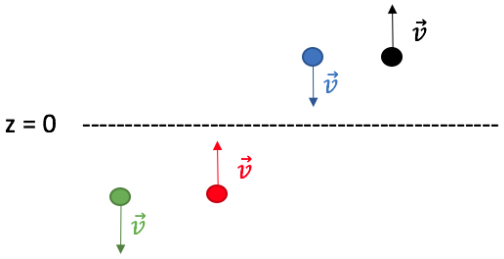
(a) $\gamma = 0.1$



(b) $\gamma = 0.5$



(c) $\gamma = 0.9$



(d) Sketch of the four zones studied for convergence

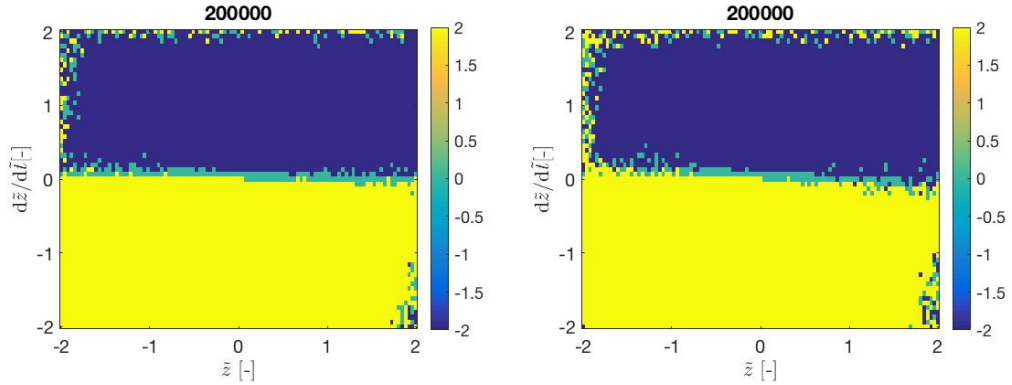
Figure 6: Evolution of the values of the Q matrix for changing discount rate

In this case, all the learning algorithms have been run for the same number of episodes because of the applied learning rate. Figure 6 shows that all the learning algorithms have converged. The smaller the discount rate, the faster the convergence of the four studied zones. Once again, the quality of the learning process cannot be concluded from this figure, but applying the exploitation algorithm.

One particular issue while reducing the discount rate is that the horizontal lines of convergence get closer. In other words, the absolute values of convergence of the different zones are closer the smaller the discount rate. It has been said in Section 3.3 that the absolute value of convergence of each zone can be related in some way

to how easy it is for the vehicle to reach the desired state when it is found in that zone. It seems that when the discount rate is reduced, meaning that the value of the future state is less important, the autopilot misses this kind of intuition. It is no longer able to predict at what zone the vehicle will be closer to the goal state. The only prediction it can make at $\mu = 0.1$ is that the green zone is the worst scenario the vehicle can find in order to reach the desired state.

Some conclusions from the previous paragraph can be observed in Figure 7. As the discount rate decreases, the action to take at each state is more dependent on the speed of the MAV. Therefore, the green line that separates the yellow and the blue zones tends to get horizontal, separating the positive from the negative speed states. It seems that the policy map gets more rudimentary. Actually, as it was said in Section 2.2 the discount rate establishes the weight of the future state against the weight of the present state in the update of the policy map. That is why for a lower discount rate the learning algorithm misses future prediction since it has updated the Q matrix giving more importance to the present state. If prediction is missed, it would be normal to think that every time the MAV is falling down the best thing to do is to apply maximum thrust up and if it is going up, thrust down.



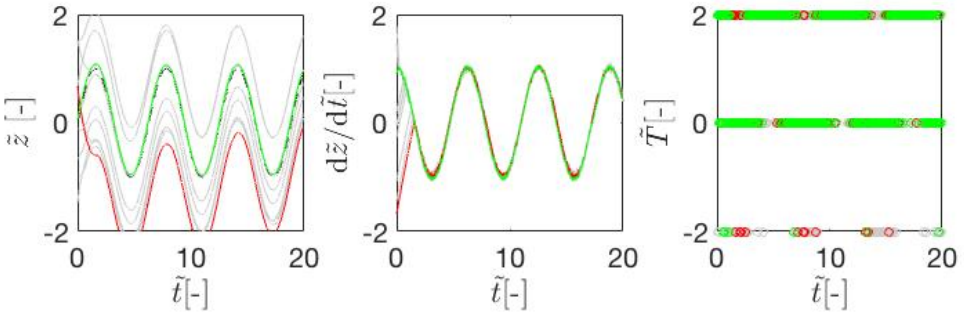
(a) $\gamma = 0.1$ (b) $\gamma = 0.5$

(c) $\gamma = 0.9$

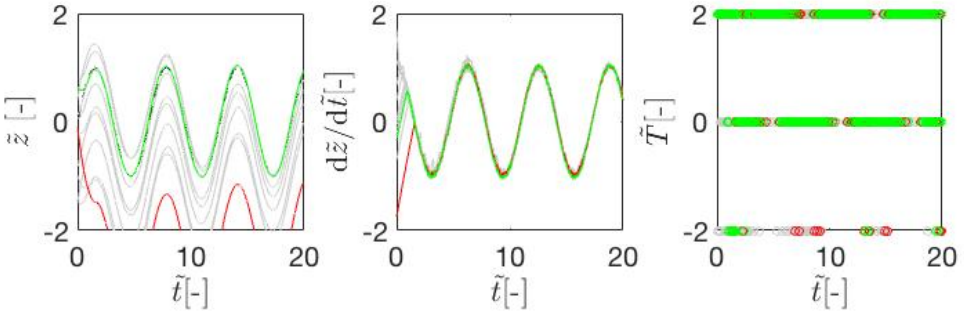
Figure 7: Representations of the policy maps for changing discount rate

These conclusions have been reflected during the exploitation algorithm. It can be

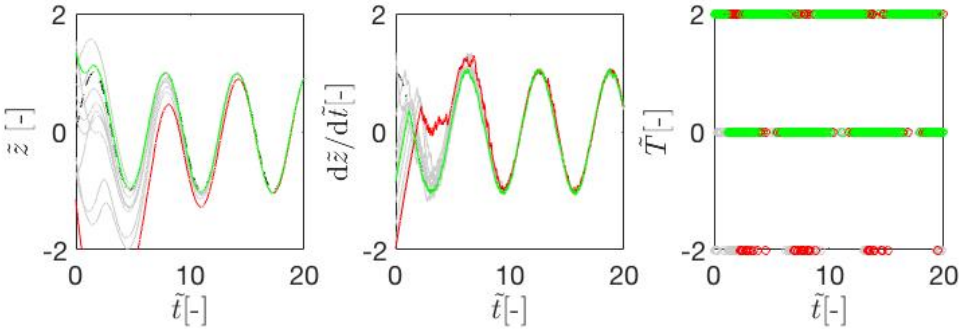
observed in Figure 8 that for the cases of 0.1 and 0.5 discount rate, the speed is stabilized much sooner than for a 0.9 discount rate. In fact, having a look again at Figure 7 for the representation of the policy maps, it can be appreciated that for the smallest value of γ the action that the policy map sets as the best one is just a function of the vehicle velocity: the best action gets independent of the position. Hence, it can be said that for a small γ the value of the ϕ parameter in the reward function becomes useless since the position of the vehicle is almost irrelevant at the time of choosing an action. At $\gamma = 0.1$, the system is only able to track the velocity. One could conclude by having a look at Figure 8 that the greatest discount rate minimizes the position stabilization period maintaining a negligible error in velocity.



(a) $\gamma = 0.1$



(b) $\gamma = 0.5$



(c) $\gamma = 0.9$

Figure 8: Evolution of position, speed and actions for changing discount rate. Simple harmonic motion. In green, the episode with the smallest position error; in red, the episode with the largest position error

Since not all the episodes that have been run in the exploitation algorithm have been plotted on Figure 8, Table 3 shows the mean position error of all episodes for each of the three different discount rates. It is easily concluded that a discount rate of 0.9 is truly the optimal value to use since all the episodes show a mean position error that allows a successful performance of the vehicle: it minimizes the mean error and it is the only case that presents zero crashes, which is in fact the most important goal.

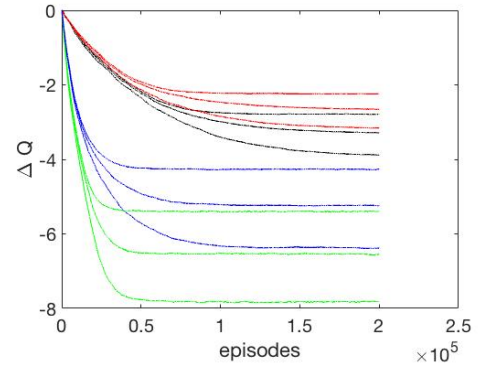
Discount rate	Error< 0.3	Error< 1	Error< 10
0.1	40.90	61.70	94.74
0.5	48.93	81.44	96.94
0.9	100	100	100

Table 3: Percentages of the number of episodes for which the mean position error after settling time is smaller than 0.3, 1 and 10 for changing discount rate. Simple harmonic motion. Total number of episodes: 10000

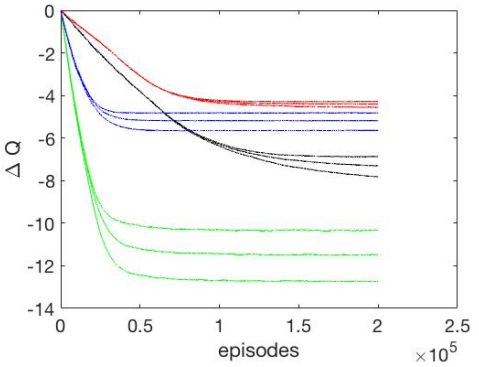
3.5 Effect of the Reward Function parameter

The only parameter that still needs to be optimized is ϕ. In order to carry on with the study, the values of the learn and discount rates haven been fixed to be the ones that have been found to be optimal ($\mu = 0.1$ and $\gamma = 0.9$). As it was previously said, the mission of the ϕ parameter is to quantitatively establish the weight of position and speed in the reward function. The values that will be included in the analysis are $\phi = 0.1$,$\phi = 0.5$ and $\phi = 0.9$.

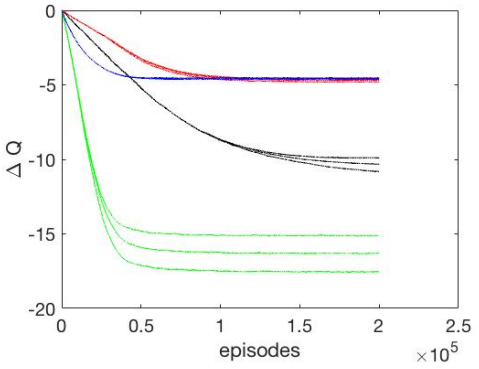
As it can be observed in Figure 9, the convergence of the policy maps for the three different values of ϕ takes place at a very similar number of episodes, meaning that the weight of position and speed in the reward function does not affect the duration of the exploration process. On the other hand, the value of ϕ shows a great influence on the absolute values of convergence of the four studied zones. The smaller weight of position in the reward function, the smaller the absolute values of convergence. As it has been observed in the policy map representations in previous sections, the choice of what action is better to take is more dependent on speed than on position. Therefore, the more concern on position, the greater the corrections that the exploration phase has to make, increasing the absolute value of convergence.



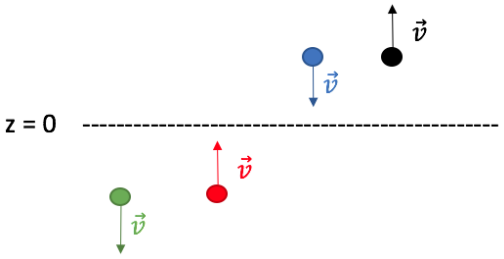
(a) $\phi = 0.1$



(b) $\phi = 0.5$



(c) $\phi = 0.9$



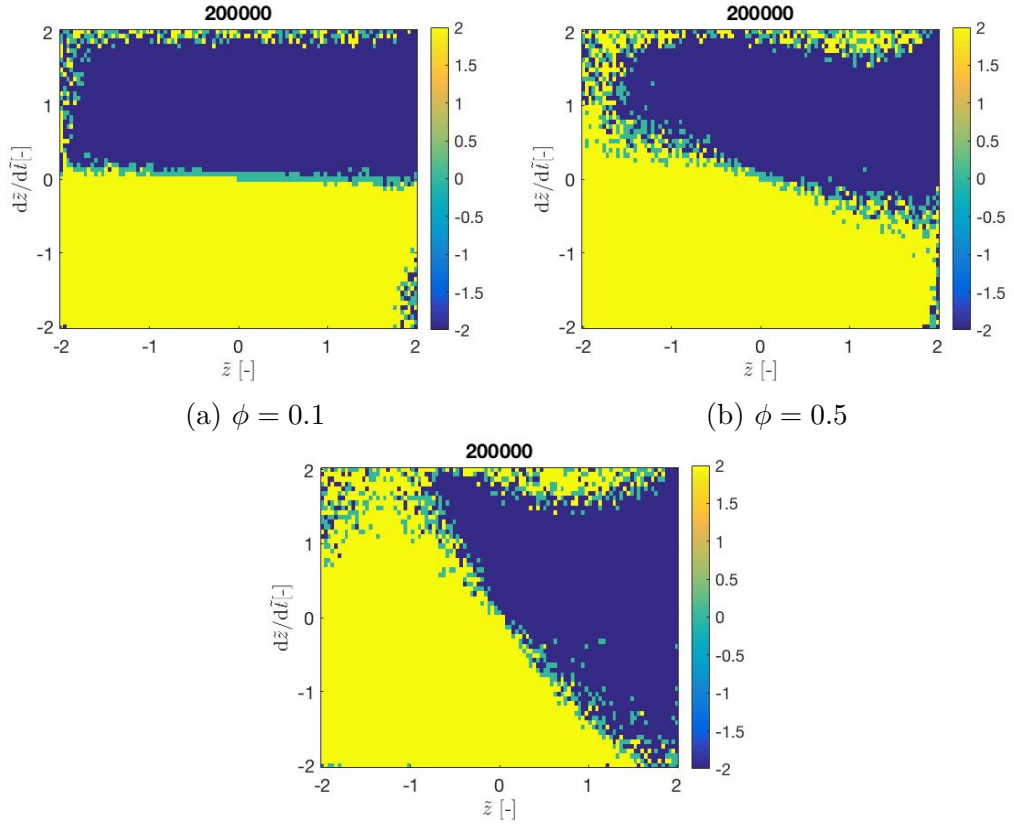
(d) Sketch of the four zones studied for convergence

Figure 9: Evolution of the values of the Q matrix for changing phi

Moreover, it is appreciated that the lines of the blue zone go upwards with respect to the rest as the ϕ parameter is increased. If more weight is given to the velocity in the reward function, there is more concern about reaching zero velocity than zero position ($\phi = 0.1$, Figure 9a). In this case, the two zones that are more likely to allow the vehicle to reach this condition are the red and black zones (where the vehicle is decelerating), so that they are upper than the green and blue zones. In the case in which the weight of position in the reward function is greater than the weight of velocity, there is more concern about reaching zero position than zero speed ($\phi = 0.9$, Figure 9c). In this case, the system considers that zero position is slightly easier to reach by the vehicle when it is at the blue zone than at the red one. This is true since at the blue zone the vehicle is accelerating downwards, reaching the zero position faster that if it is decelerating, as it is the case in the red zone.

The representations of the policy maps for the different ϕ values are shown in Figure 10. For $\phi = 0.1$, the choice on what action is the best one for the training condition is almost independent of position. It is observed that, as ϕ increases, the slope of the green line that separates the blue and yellow zones is more pronounced. It seems that although there is a very large ϕ (meaning a greater concern about position), the choice of the best action is always a function of the speed, always dependent on

velocity, since the green line never gets completely vertical.



(a) $\phi = 0.1$

(b) $\phi = 0.5$

(c) $\phi = 0.9$

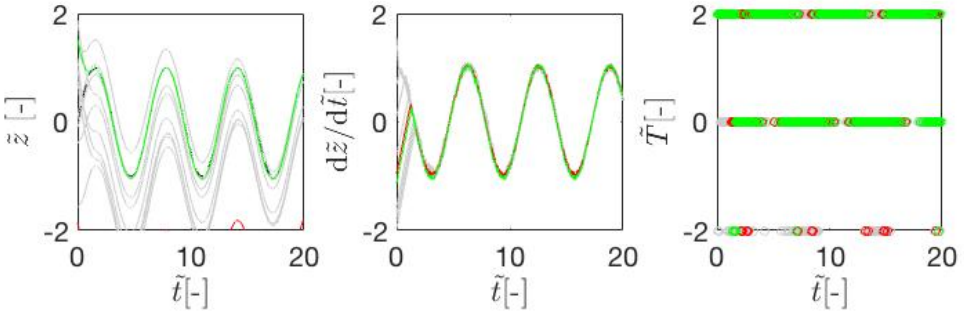
Figure 10: Representations of the policy maps for changing phi

It can be thought that showing a great concern about position may be positive in order to train the MAV. However, the application of the exploitation algorithm will vanish any doubt.

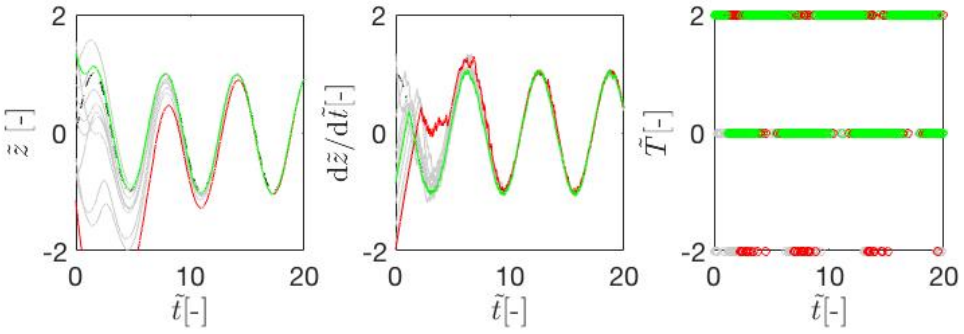
It can be appreciated in Figure 11a how the vehicle tries to follow a simple harmonic motion. If there is more concern about speed, the evolution of position seems to converge at a very late time. On the other hand, Figure 11c shows that when most of concern is placed on position, the stabilization period is minimized. The results obtained for $\phi = 0.5$ show a slightly larger position stabilization and a slightly shorter speed stabilization period than the case of $\phi = 0.9$. Therefore, Table 4 will be helpful to consider which value of phi is the optimal one.

Table 4 shows the mean position error of all the episodes that have been run for the three values of phi. It can be concluded that the case of $\phi = 0.1$ can be discarded since it is the only case for which there is possibility of crash of the MAV. However, the cases of $\phi = 0.5$ and $\phi = 0.9$ show a very slight difference. The final decision has been made on choosing $\phi = 0.5$ as it is the case for which the mean position error is minimized according to the table.

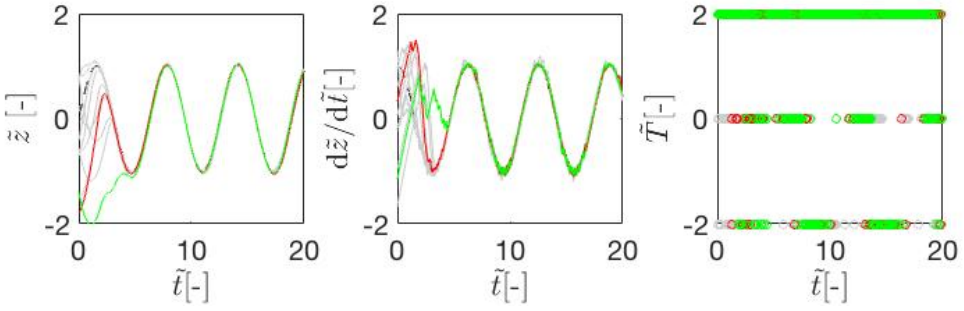
In conclusion, for further investigation in later sections the optimal set of parameters used will be $\mu = 0.1$, $\gamma = 0.9$ and $\phi = 0.5$. This is the combination that has shown the best performance for all the study. Not only does it allow the UAV to reach a target trajectory but there is no sign of crash in the simulations and the position error is minimized.



(a) $\phi = 0.1$



(b) $\phi = 0.5$



(c) $\phi = 0.9$

Figure 11: Evolution of position, speed and actions for changing ϕ. Simple harmonic motion. In green, the episode with the smallest position error; in red, the episode with the largest position error

phi	Error< 0.3	Error< 1	Error< 10
0.1	43.92	71.73	96.46
0.5	100	100	100
0.9	99.82	99.94	100

Table 4: Percentages of the number of episodes for which the mean position error after eight seconds is below 0.3, 1 and 10 for changing phi. Simple harmonic motion. Total number of episodes: 10000

3.6 Resolution

Now that the parameters of the Q-Learning algorithm have been established, it would be interesting to study the impact of the resolution of the state space on the learning process. This analysis will be useful if more degrees of freedom are introduced into the problem since the increasing size of the Q matrix may not be affordable for the CPU. For the previous simulations, the position was discretized into 103 elements and speed, into 63 elements. This combination showed consistent results.

Case	Resolution	Error< 0.3	Error< 1	Error< 2	Error< 10
Case 1	103x63	100	100	100	100
Case 2	60x40	97	99.44	100	100
Case 3	20x10	61.75	61.75	61.75	92
Case 4	10x10	9	89	100	100

Table 5: Percentages of the number of episodes for which the mean position error after settling time is under 10, 2, 1 and 0.3, respectively. Simple harmonic motion. Resolution is [axb], being a the elements of discretisation of position and b, of speed. Total number of episodes: 10000

As Table 5 shows, four different resolutions have been simulated for the already defined algorithm parameters. The number of episodes that have been simulated for the exploration process has been dependent on the resolution of each case since the rougher the resolution, the shorter the required training process. A resolution of 103x63 has been shown to need a number of episodes close to 200000 to converge since a larger Q matrix implies a longer exploration period.

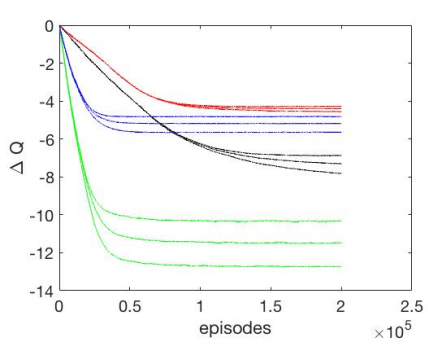
It can be appreciated in Table 5 that reducing the number of elements of discretization does not necessarily mean obtaining inconsistent results. However, for the two most robust resolutions (cases 3 and 4), some conclusions can be reached regarding the proportion between the elements of the discretization of position and speed.

When resolution is reduced, it seems that for $a/b = 2$ (case 2), the worst results are obtained since for the 92% of the episodes the MAV trajectory ends in a crash.

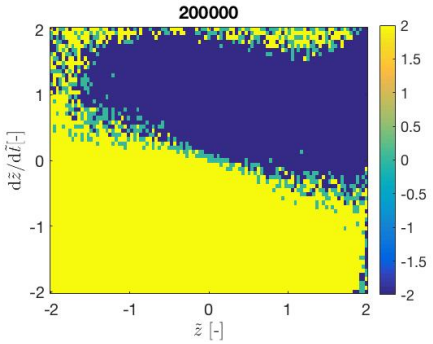
While that ratio tends to unity (case 4), results get more consistent. For Case 4 there is no episode in which the MAV crashes although the state space is discretized into a smaller number of elements. Furthermore, the performance of the vehicle in Case 4 is very satisfactory since the mean error of all the episodes is smaller than two and a 89% shows a mean position error smaller than one. This means that the performance of the set of algorithms is not drastically deteriorated although the description of the training state space is much more robust.

Having a look at Figures 12a and 12c it can be observed that their respective policy maps have clearly converged, since the colour lines get horizontal and steady. Figure 12e shows that for Case 3 the colour lines oscillate about a horizontal asymptote and it does not seem to converge further, it is not likely to get steady. The most particular case is Case 4, because it is the only case for which the three colour lines that correspond to each of the zones are all overlapped.

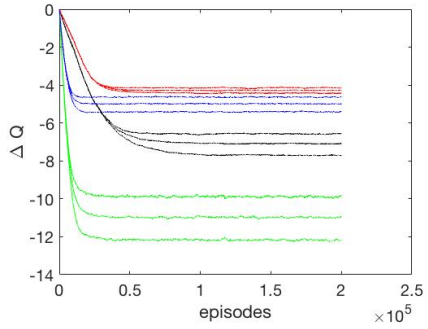
The shape of the policy maps is very similar for all cases: the inclined green line separates the upper yellow zone from the bottom blue zone. The policy map of Case 4 is the most robust map. However, according to Table 5 it is not the worst policy map. In conclusion, when resolution is reduced, maintaining a resolution proportion of unity (same resolution for speed and position) the quality of the exploration and exploitation algorithms is limited, but not missed.



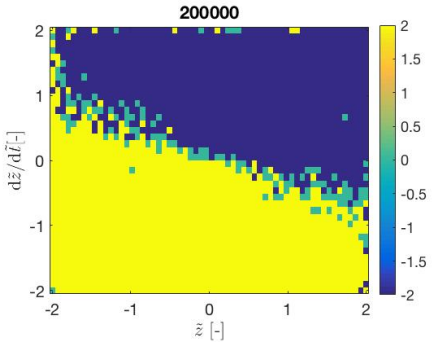
(a) 103x63 - Q evolution



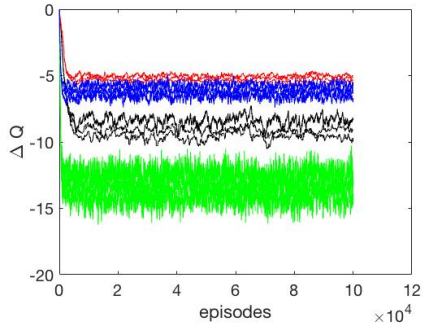
(b) 103x63 - policy map



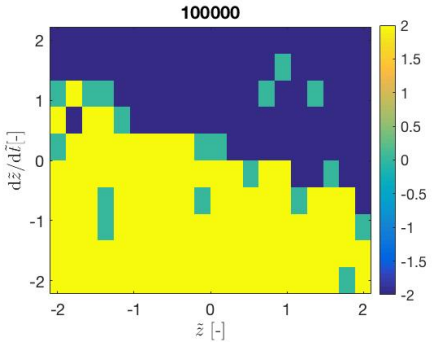
(c) 60x40 - Q evolution



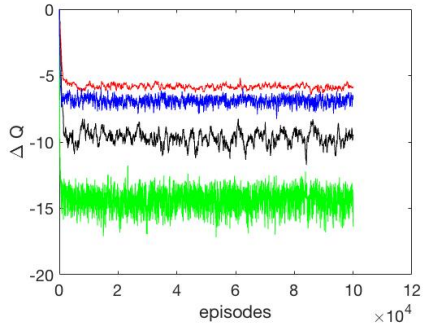
(d) 60x40 - policy map



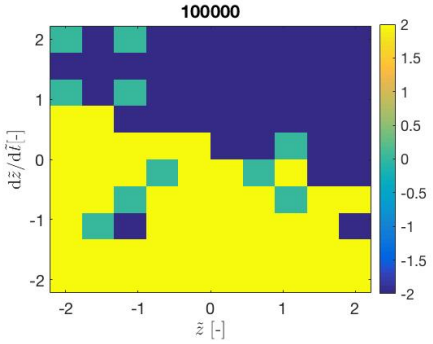
(e) 20x10 - Q evolution



(f) 20x10 - policy map



(g) 10x10 - Q evolution



(h) 10x10 - policy map

Figure 12: Evolution of the mean values of the Q matrix and policy maps for changing resolution

4 Two Degrees of Freedom Problem

The 'toy problem' was merely an example for the reader to understand the Q-learning algorithm and how its parameters affect the performance of the learning process. It has also been useful to set the values of those parameters in order to optimize the exploration performance. The problem was defined as non-dimensional and the characteristics of the vehicle were roughly approximated because, from an engineering point of view, it is more interesting to aim for total longitudinal control. In a more real case, someone would like to control a vehicle that moves at least in a plane, in a motion of two degrees of freedom.

The case of the motion on a horizontal plane has not been analyzed in this project since it is a very simple situation. It is more interesting from an engineering point of view to study the control of the movement of the MAV in the vertical plane (XZ plane), where gravity is also playing a role. As it will be later confirmed, the control in the horizontal axis does not become a challenge, but the control in the vertical direction will impose some limitations on the performance of the MAV.

In this chapter, a more realistic vehicle will be simulated and the performance in the vertical plane of the MAV will be analyzed attending to the desired performance for an engineering application. Some aspects that will be studied in this section are the valid range of speeds for the performance of the MAV, the dynamics limitations, the validity of a unique policy map for different vehicles, the performance of the MAV following different types of trajectories and the effect of the reduction of the state space covered in the policy map. A rough power consumption analysis will be carried out in an attempt to estimate the energy consumption of a flapping wing vehicle.

4.1 Problem definition

As it has been previously said, for an engineering application it is more convenient to control the vehicle motion at least in two dimensions. In this project and seeking for longitudinal control, the plane of motion for this two degrees of freedom case has been chosen to be the XZ plane. The fact that the MAV can move vertically will impose limitations on its performance since gravity is the only force that can push the vehicle downwards, but this issue will be commented later in more detail. Therefore, in this two dimensional control case, the vehicle can move in space with two degrees of freedom, which are x and z according to the dynamic model shown on Figure 13. Aerodynamic drag, weight and thrust are the responsible forces for the motion of the vehicle.

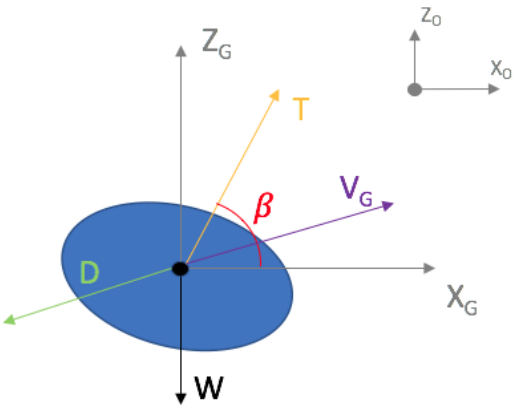


Figure 13: Dynamic Model for a two degrees of freedom motion

In a two dimensional scenario, the thrust needs to be projected onto the x and z axis in order to allow a complete movement in space. When it comes to a flapping wing vehicle, this change of direction of the thrust vector is possible thanks to the stroke plane angle β. The stroke plane is the plane on which the flapping wings rotate. In this problem, β is defined as the angle between the normal vector of the stroke plane, which coincides with the direction of thrust, and the horizontal direction. This parameter will be used as a control input. In real life, insects an birds have limited stroke plane angle possibilities since their bodies show physical limitations. In fact, in the case of an MAV, as far as it is an artificial device, these limits are imposed by the designer depending on the motion the MAV needs for its application.

Newton's law is now applied in the two directions of motion (Eqs. 11 and 12)

$$m\frac{d\dot{x}}{dt} = \sum F_x \tag{11}$$

$$m\frac{d\dot{z}}{dt} = \sum F_z \tag{12}$$

Applying the dynamic model shown on Figure 13, the general equations of motion are derived

$$m\frac{d\dot{x_G}}{dt} = T\cos\beta - \frac{1}{2}\rho S V_G^2 C_D(\frac{\dot{x_G}}{|V_G|}) \tag{13}$$

$$m\frac{d\dot{z_G}}{dt} = T\sin\beta - mg - \frac{1}{2}\rho S V_G^2 C_D(\frac{\dot{z_G}}{|V_G|}) \tag{14}$$

where T is the module of thrust, g corresponds to gravity, ρ is air density, m accounts for the mass of the vehicle, V_G is the velocity of the vehicle, S is the wing surface and C_D accounts for the drag coefficient.

Note that in the equations of motion, the velocity of the vehicle is expressed as $V_G = \sqrt{\dot{x_G}^2 + \dot{z_G}^2}$ and it is coincident with the total aerodynamic velocity since the

wind flow has not been taken into consideration. Differently to [9], in this project the effect of the flow conditions is never considered. As the name of the project indicates, the objective here is to design a preliminary autopilot. Once through this machine learning method the designer gets a first approach of the policy map for the application of the MAV, some wind tunnel experiments may be carried out in order to include wind effect into the reinforcement learning process. The next step in the design of the final autopilot would be something similar to the learning process of perched landing carried out by Bristol University in [5].

Once again, in the design of the autopilot the exploration and exploitation algorithms will be isolated and they will be applied following the same process as in the one dimensional case. Therefore, in the same way, the reward function (Eq. 15) needs to be established in order for the MAV to know how far it is from reaching its goal, the target position:

$$R(t) = -\phi(|x_G(t)-x_0(t)|^2+|z_G(t)-z_0(t)|^2)-(1-\phi)(|\dot{x_G}(t)-\dot{x_0}(t)|^2+|\dot{z_G}(t)-\dot{z_0}(t)|^2) \tag{15}$$

In the two d.o.f. case, the learning algorithm will be training the MAV to reach the $x_0(t) = 0, z_0(t) = 0, \dot{x_0}(t) = 0, \dot{z_0}(t) = 0$ state, the equivalent with respect to the one dimensional case. As it can be appreciated in Eq. 15, the ϕ parameter is also used in the same way, establishing the weight difference between position and speed inside the reward function.

4.2 Definition of the vehicle and conditions

Once the dynamics of the problem have been defined, it is time to establish the characteristics of the proposed vehicle and its environment. Of course, the vehicle has to be defined by the designer although in this project a first approach is going to be made trying to simulate an engineering application.

Regarding the environment conditions, the MAV is going to be assumed to fly at Sea Level conditions, meaning that the gravity and density of air will adopt the following values

- $g = 9.81 m/s^2$
- $\rho = 1.225 kg/m^3$

In order to set the first model for the MAV, the flapping wing vehicle defined in [10] will be useful to have a closer description of the physical characteristics of this kind of vehicle. In the proposed article, the authors have designed and evaluated a LIPCA-actuated flapping wing device. A LIPCA is a Lightweight Piezo-composite actuator. The flapping wing device used in the experiment weights $20g$ and has a wing surface of $29cm^2$, which leads to a wing loading of approximately $6.897 kg/m^2$. This is the relationship between mass and surface that will be used in all the following sections.

A light-weight vehicle will be design as a first model, which will result in a small device that could be useful in a visual inspection of difficult accessibility. Think of a device with a weight similar to the one of a cell phone. For this case, the vehicle will have the following properties

- $m = 0.1kg$
- $S = 0.0145m^2$

Moreover, as a flying object, the MAV is going to experience some aerodynamic drag, so the drag coefficient will be assumed to be the one of a sphere at a Reynolds number of 3400 according to the theoretical results in [11], so that

- $C_D = 0.44$

A very deterministic characteristic of the vehicle with regard to its future performance is the frequency at which a new action can be taken, in other words, the frequency of reaction. Again, the assignment of value for this property has been taken considering the experiments in [10]. It has been considered that in an ideal case (remember that this is a preliminary design of an autopilot) the MAV will be able to change the direction of the thrust as soon as it can perform a new stroke, so that the reaction frequency will be associated to the flapping frequency. Although the module of the thrust should also be dependent on the flapping frequency, the current algorithms do not make it possible to change the frequency of taking an action during the performance, so the frequency of reaction will be assumed constant and the module of thrust will be set according to the exploration and exploitation algorithms criteria.

In order to obtain the results of the experiment in [10], the range of flapping frequencies was varied from 4 to 15 Hertz. As a result, they obtained that the frequency that produced the greatest forces was coincident with the natural frequency of the device: 9 Hertz for their case. Therefore, the optimum frequency depends on the structural design of the device. Since the simulations of this Bachelor's Thesis are not applied on an already existing device, there is no natural frequency to be measured, so that the frequency of taking an action will be assumed to be equal to 10 Hertz.

As it has been said in the previous section, this vehicle has two control inputs: the module of thrust and the stroke plane angle. The range of values the module of thrust can adopt is maximum thrust, half of the maximum thrust and zero thrust. This time thrust will never point downwards, differently from the one dimensional case. The only force that will make the MAV go down is gravity. The value of maximum thrust will be set so that $T_{max} \approx 2mg$, making it possible to perform a large number of maneuvers

- $T_{max} = 2N$

In the case of the stroke plane angle β, the set of possible options has been set to include 60, 90 and 120 degrees according to its definition (See Figure 13). Since the

critical direction for control is the vertical one, this set of angles has been chosen so that the projection of thrust on the vertical axis is larger than in the horizontal one. The combination of the possible modules and directions established for the thrust force allow a total of nine different actions for the two degrees of freedom problem. Remember that the number of columns of the Q matrix is equal to the number of available actions. Therefore, the number of elements of the Q matrix is directly proportional to the number of possible actions. Then, it is convenient to find a set of actions that maximizes manoeuvrability at the same time that the number of possible actions is minimized.

In this two dimensional case, considering an open-air application, the chosen range for position will be equivalent to the one dimensional case since it allowed satisfactory results and, if looking for accuracy the range is reduced, there is more probability of finding the MAV out of the position and speed ranges. Take into account that the consequences of this situation are still unknown, so there is concern about redundancy at this moment. Remember that all the points that are found out of either the speed or position range are all treated as its closest point found in the policy map, which may not be specific for that state. The established ranges for the state space have been fixed in advance, but further sections will confirm that it is a reasonable choice

- $-2 \leq x \leq 2m$
- $-2 \leq z \leq 2m$

The range of speed in the x and z directions will also be equivalent to the one dimensional case for the same reason

- $-2 \leq \dot{x} \leq 2m/s$
- $-2 \leq \dot{z} \leq 2m/s$

Regarding the resolution applied in this case, since two more state variables have been introduced, the size of the Q matrix has been considerably increased. In case the same resolution of the one d.o.f. problem is applied in this new problem, the number of cells of the policy map gets 6489 times larger: about 42 million rows compared to the 6489 (103x63) rows in the one dimensional case. This cipher is unaffordable for the CPU memory. Considering the results obtained in Section 3.6, it was concluded that resolution can be reduced without presenting a critical damage for the learning performance. Therefore, the resolution applied in this problem is set to be 30x30x30x30, maintaining the same size for the steps of all variables. The Q matrix of the two dimensional autopilot will therefore have 810000 rows, corresponding to the total number of possible states that the policy map includes, and 9 columns, being equal to the total number of actions that the vehicle can perform.

The time-step size used for the integration of the equations of motion will be equal to $h = 0.05s$. This value will provide accurate results since the one and two d.o.f.

problems are equivalent. According to the results in Appendix A, this time-step should be sufficiently small in order to obtain a numerical solution that is close enough to the analytic one .

4.3 Learning process

All the parametric study carried in the one dimensional case (Section 3) will be considered in this section. The learning algorithm parameters will be freezed to those that were found to optimize the one dimensional learning:

- $\mu = 0.1$
- $\gamma = 0.9$
- $\phi = 0.5$

After the introduction of two new state variables (x,$\dot{x}$) and a new control input (the stroke plane angle β), the number of dimensions of the policy map has considerably increased. Therefore, the policy map in this two dimensional control study will not be represented since it becomes a very difficult task. That was one of the main reasons behind the study of the one dimensional case, being its policy map very simple to represent and intuitive to interpret. This made it easier to explain and understand the concept of a policy map.

However, it is still important to be aware of the convergence of the policy map and it can be done in the same way as in the one dimensional case. In the two dimensional movement, the convergence of a total of sixteen zones will be studied depending on the sign of each state variable (2^4). In this case, each zone will be identified according to the following colour criteria, that will allow an easier interpretation. Depending on the sign of each of the state variables, a percentage of red, blue and green will be assigned to each zone according to Tables 6, 7 and 8.

$x > 0$ $z > 0$	$x > 0$ $z < 0$	$x < 0$ $z > 0$	$x < 0$ $z < 0$
100%	75%	50%	25%

Table 6: Percentages of red color assigned to the position state with respect to the target reference frame.

$\dot{x} > 0$	$\dot{x} < 0$
100%	0%

Table 7: Percentages of green color assigned to the sign of the horizontal velocity with respect to the target reference frame.

$\dot{z} > 0$	$\dot{z} < 0$
100%	0%

Table 8: Percentages of blue color assigned to the sign of the vertical velocity with respect to the target reference frame.

In this way, Figure 14 shows the evolution of the mean values of the sixteen zones that have been studied. At first sight it can be observed that all zones are converged since all the colour lines get horizontal and that the number of episodes for total convergence is very large compared to the one dimensional case. The total number of episodes in this simulation has been 85 million.

The increase in the number of episodes needed for convergence is the cause of a greater absolute value in the cells of the policy map. This means that the number of corrections needed for the complete training of the vehicle has been very large compared to the one d.o.f. problem, which was expected due to the size of the new Q matrix and the increasing complexity of the problem. Moreover, the absolute value associated to each zone cannot be now linked to how easy it is for the vehicle to naturally reach the goal state at each zone since the learning process is now much more complex. What can be concluded for sure is that the upper green and yellow zones are the ones that converge first since they imply that the MAV has a negative vertical speed. Once again, the first thing that the MAV learns is not to fall down.

However, the zones that imply a positive vertical speed are more difficult to reach. These zones start to converge after the MAV has learned how not to fall, which makes sense according to the conclusions of the one d.o.f. problem. Since now there is a much wider position range and thrust is projected onto two directions, the MAV needs many more episodes to train in these zones, where the best action to take may not be a straight forward decision.

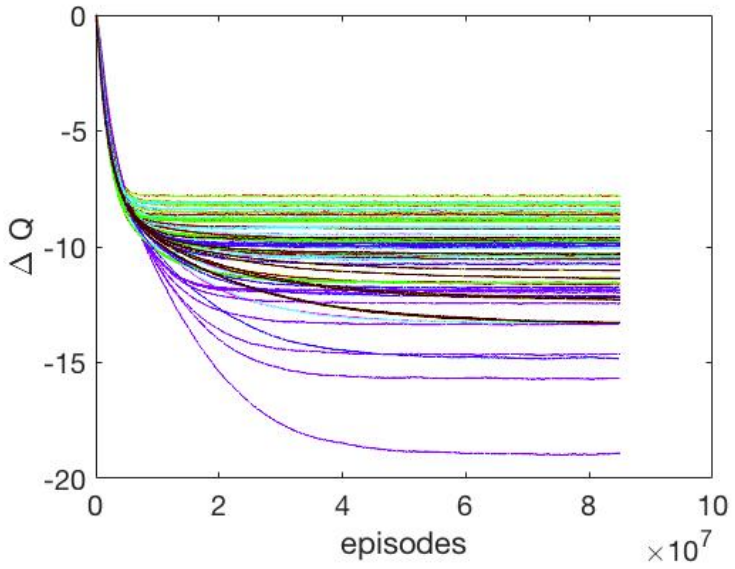


Figure 14: Evolution of the mean values of the Q matrix in the two d.o.f. study

4.4 Characteristic velocities

Before starting to use the exploitation algorithm, it would be appropriate to determine in a first approximation what will be the limitations that the proposed vehicle will find according to its definition. Probably, the limitations will be more restrictive when the exploitation algorithm is simulated.

The discrepancy between theoretical calculations and the real performance of the vehicle in the simulations might be mainly caused due to the rudimentary set of actions that the MAV can perform, the range of position and speed included in the policy map and the resolution of the state space.

4.4.1 Terminal velocity

By definition, when an object is falling down just by the action of gravity (free-fall), the speed that the falling object reaches when the drag force equals weight is known as the terminal velocity. Once this speed is reached, the falling object is no longer accelerated. Since in this problem the only force that pushes the vehicle downwards is gravity, the terminal velocity can be interpreted as the maximum downward vertical speed that the MAV can afford. According to Newton's second Law (Eq. 14) and assuming that the vertical velocity is equal in module and opposite in direction to the total velocity

$$m\frac{d\dot{z_G}}{dt} = 0 = -mg + \frac{1}{2}\rho S V_G^2 C_D \tag{16}$$

$$V_{Terminal} = \sqrt{\frac{2mg}{\rho S C_D}} \tag{17}$$

Introducing the design properties assigned to the vehicle and Sea Level conditions into Equation 17, a terminal velocity of $V_{terminal} = -15.84m/s$ is obtained. It has to be taken into account that the terminal speed needs some time to be reached, so that it is not a velocity that can be instantaneously achieved.

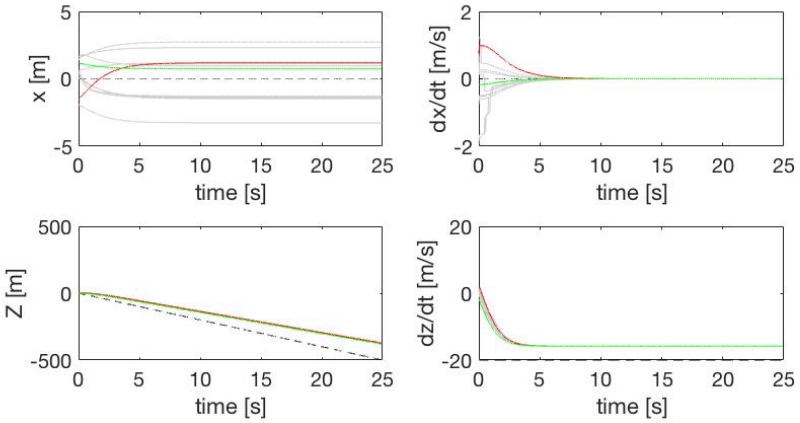


Figure 15: Performance of the MAV when a target $V_z = -20m/s$ is imposed. In green, the episode with the smallest position error; in red, the episode with the largest position error

It can be observed in Figure 15 that when a negative vertical velocity of $20m/s$ is set as target velocity, the MAV is not able to reach it: it stands at the theoretical terminal velocity of $V_z = -15.84m/s$. In order to go down as fast as possible, the only thing that the vehicle can do is to use zero thrust. It is appreciated that the vehicle considers that the error in the horizontal position is not that important in this case. In fact, according to the given set of actions, if it tries to correct the error in the x position, it will dramatically penalize the error in the vertical position and speed since the vertical projection of thrust is very large.

4.4.2 Maximum upward vertical velocity

Another limit that is going to affect performance and that depends on the characteristics of the vehicle is the maximum positive speed in the vertical axis. Consider that this velocity will be reached applying maximum thrust in the vertical direction ($T = 2N, \beta = 90^{\text{o}}$). Applying Equation 14 to a vertical movement at maximum vertical velocity condition gives

$$m\frac{d\dot{z_G}}{dt} = 0 = T_{max} - mg - \frac{1}{2}\rho S V_G^2 C_D \tag{18}$$

$$V_{z,max} = \sqrt{\frac{2(T_{max} - mg)}{\rho S C_D}} \tag{19}$$

Once the vehicle design characteristics and Sea Level conditions have been introduced into Equation 19, the maximum vertical speed is found to be $V_{z,max} = 16.15 m/s$.

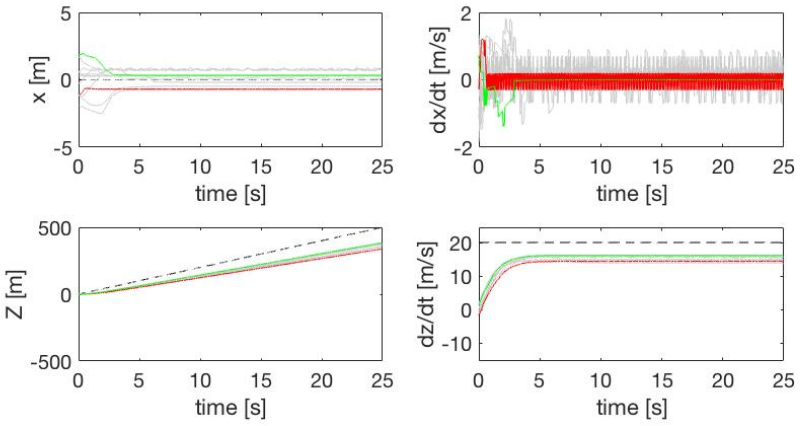


Figure 16: Performance of the MAV when a target $V_z = +20 m/s$ is imposed. In green, the episode with the smallest position error; in red, the episode with the largest position error

When a positive vertical speed of $20 m/s$ is set as the target trajectory, the MAV has no doubt that it has to apply maximum thrust. As it is shown on Figure 16, in this case the MAV considers that it can slightly correct its position in the x direction at the beginning of the performance although there is a moment at which it just tries to maintain a constant (or oscillating) position in x and care about correcting the rest of the state variables. The maximum vertical velocity reached by the MAV in the performance is exactly $V_z = 16.15 m/s$, which corresponds to the episode that shows the best performance on Figure 15 (represented with green lines). However, due to the oscillating behaviour that the MAV shows at some episodes, this maximum velocity is not always reached. In order to correct the error in the x position, the thrust is not entirely projected on the vertical axis, so that the maximum vertical acceleration is not reached. The range of maximum vertical velocities that the MAV shows goes from 14 to 16.15 m/s.

4.4.3 Maximum horizontal velocity

The maximum horizontal velocity for the designed vehicle is going to be reached when the maximum projection of the maximum thrust over the x axis is found. Let's consider the general equilibrium equations derived from Newton's Law (Eqs. 13 and 14)

$$m\frac{d\dot{x_G}}{dt} = 0 = T\cos\beta - \frac{1}{2}\rho S(V_x^2 + V_z^2)C_D\frac{V_x}{\sqrt{V_x^2 + V_z^2}} \tag{20}$$

$$m\frac{d\dot{z_G}}{dt} = 0 = T\sin\beta - mg - \frac{1}{2}\rho S(V_x^2 + V_z^2)C_D\frac{V_z}{\sqrt{V_x^2 + V_z^2}} \tag{21}$$

If there were no constraints on the applied action, it would be enough to apply maximum thrust and assume a pure horizontal motion in order to obtain the maximum horizontal speed that the vehicle can afford and the stroke plane angle at which this is possible. Therefore, introducing $V_z = 0$, $T = 2N$ and the environmental conditions of the problem into Equations 20 and 21, the maximum horizontal speed would be achieved at a stroke plane angle of $\beta = 29.36^{\text{o}}$, reaching a value of $V_{x,max} = 21.12m/s$.

For the set of stroke plane angles considered in this problem, the maximum projection of thrust on the horizontal axis will take place for either $\beta = 60^{\text{o}}$ and $\beta = 120^{\text{o}}$ at maximum thrust. Therefore, introducing this condition ($T = 2N$, $\beta = 60^{\text{o}}$) into Equations 20 and 21 according to the environmental design conditions, the maximum possible horizontal velocity is $V_{x,max} = 14.30m/s$ at a vertical velocity of $V_z = 10.74m/s$. This result is leading to the conclusion that the performance of the MAV is going to be poor at the time of describing a horizontal line at a horizontal speed close to the maximum due to the limited set of actions.

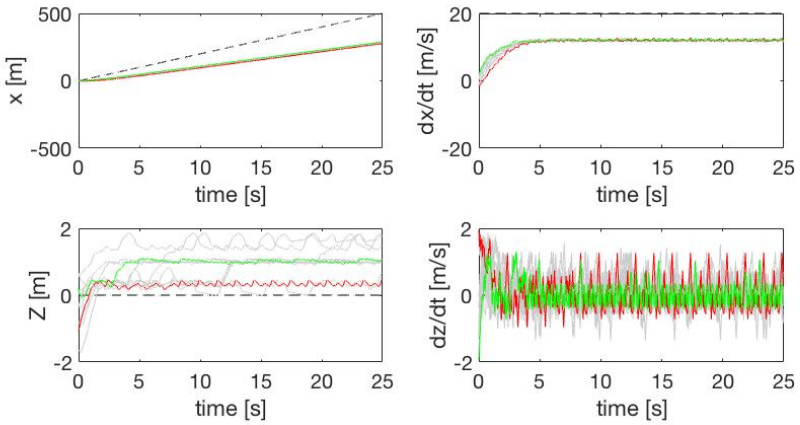


Figure 17: Performance of the MAV when a target $V_x = +21m/s$ is imposed. In green, the episode with the smallest position error; in red, the episode with the largest position error

According to Figure 17, predictions have been confirmed. When a target horizontal velocity of $20m/s$ is set, the MAV is not able to follow a straight line trajectory because of the reduced set of actions it can perform. If it applies a 2 Newtons thrust at a stroke plane angle of 60 degrees, it accelerates in the horizontal direction but also in the vertical one, so it has to correct the trajectory at almost every time it takes an action. There may be some moments at which the MAV applies zero or one Newton thrust, which means that the acceleration in the horizontal direction is not always equal to the maximum. This is the reason why the maximum horizontal velocity reached is about $12.2m/s$, which is much lower than the maximum horizontal velocity that the maximum thrust of the vehicle can reach according to Equations 20 and 21.

4.5 Performance

The best autopilot and the best vehicle on Earth would include no limitations in the set of actions to take. However, this Bachelor's Thesis is about the preliminary design of an autopilot and it shows some limitations. The number of possible actions considerably affects the size of the Q matrix, which is finite due to the CPU RAM memory. The consequence of establishing a rough set of actions can be noticeable while performing some trajectories, narrowing the range of trajectories that the MAV can afford and, therefore, its manoeuvrability.

As it has been said before, there is only one force that is going to push the MAV down: gravity. Consequently, at some trajectories the downward acceleration reached may not be enough. Starting from rest, the maximum vertical acceleration that the MAV can reach while it is going down is $9.81m/s^2$. This can limit the performance of the MAV if a faster movement is desired.

In this section, different types of trajectories will be performed by the MAV in order to check the effectiveness of the training process and the imposed set of actions.

4.5.1 Uniform Circular Motion

In a uniform circular motion, the centrifugal force needed at the upper point of the circle can be critical. Imagine a circular target trajectory of constant angular speed ω and radii R (See Figure 18). According to the given set of actions, it is going to be easier for the MAV to control motion upwards. At first sight, the maximum negative acceleration in the vertical direction is going to be reached at point A, which can become a critical point. Furthermore, point B will also be analyzed since the MAV is found at negative vertical velocity, which may also become difficult for the vehicle to control.

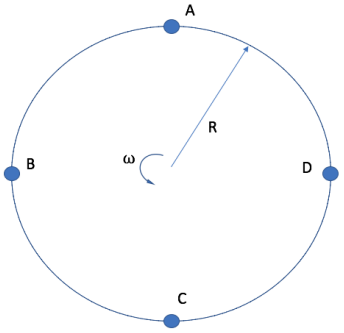


Figure 18: Uniform Circular Motion

At point A, the vertical velocity component is zero ($V_z = 0$), the horizontal component of velocity is equal to the module of speed ($V_x = \omega R$), tangential acceleration is zero ($\ddot{x} = 0$) and vertical acceleration is maximum and equal to the normal acceleration. At this position, the action that the MAV uses is zero thrust, which is reasonable, since it is the only way to push the MAV down. In the horizontal motion, the module of the drag is very small, so that the tangential acceleration is almost negligible. However, the application of Newton's Law in the vertical direction can impose a constraint

$$m\frac{d\dot{z_G}}{dt} = -m\omega^2 R = -mg \tag{22}$$

As it is shown in Equation 22, in order for the MAV not to get away from the trajectory

$$\omega^2 R \leq g \tag{23}$$

Looking at Figure 19, it can be observed that when the MAV is told to follow a moving target (discontinuous black line) describing a circle of twelve meter radii

with an angular velocity of one radian per second ($\omega^2 R = 12$), the actual trajectory diverges from the target one in all cases. Predictions have been confirmed. Not only does the z motion diverge from the target position, but the amplitude of both the x and z motions are smaller than the target one and irregular.

After this demonstration, one would think that it would be enough to satisfy the $\omega^2 R \leq 9.81$ condition in order for the MAV to perform a successful trajectory. However, having a look at Figure 20, for the case of a trajectory of $\omega = 1rad/s$ and $R = 8m$ it seems that the actual trajectory is not uniform although it attempts to reduce the error at some points.

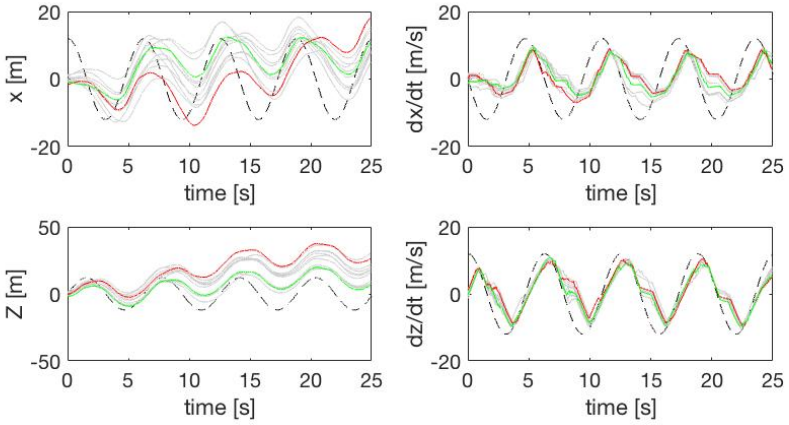


Figure 19: Uniform Circular Motion performance for $\omega = 1rad/s, R = 12m$. In green, the episode with the smallest position error; in red, the episode with the largest position error

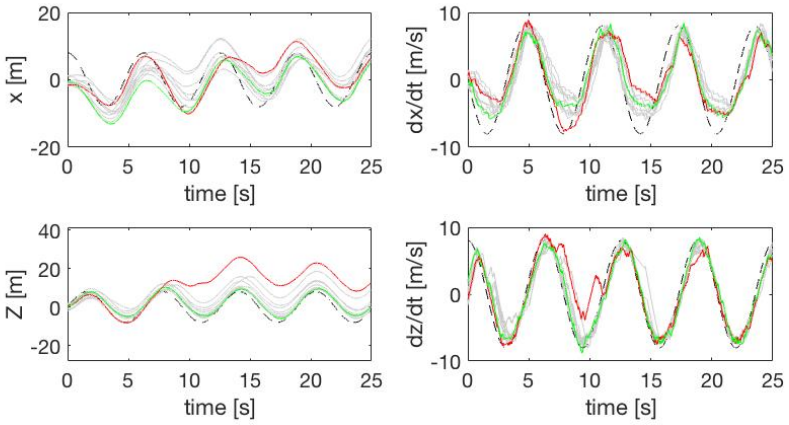


Figure 20: Uniform Circular Motion performance for $\omega = 1rad/s, R = 8m$. In green, the episode with the smallest position error; in red, the episode with the largest position error

The reason for this non-uniform performance can be explained with the analysis of the behavior of the MAV when it is found at point B of the UCM (Fig. 18). At this point, the horizontal component of velocity is zero ($V_x = 0$), the vertical velocity is negative and its module is equal to the total velocity ($V_z = \omega R$), the tangential acceleration is zero ($\ddot{z} = 0$) and the horizontal acceleration is equal to the normal acceleration. At point B, in order to keep zero tangential acceleration, the module of thrust is set to one Newton. In order to have some centrifugal force in the horizontal direction, the stroke plane angle is set by the Q matrix to 60 degrees. Applying this action ($T = 1N, \beta = 60^{\text{o}}$), there is no problem to reach the zero tangential acceleration, but the implementation of Newton's Law in the horizontal direction leads to the following relationship

$$m\frac{d\dot{x_G}}{dt} = m\omega^2 R = T\cos\beta \tag{24}$$

which leads to the conclusion that in order to afford the required normal acceleration it is needed to satisfy the following condition

$$\omega^2 R \leq \frac{T\cos\beta}{m} = 5 \tag{25}$$

This is an approximation. It is shown in Figure 21 that, at the limit condition, the performance of the MAV is almost equal to the target one. As it can be concluded, if the stroke plane angle had a wider set of configurations, the limiting $\omega^2 R$ factor could be increased, so that the MAV could broaden the range of affordable trajectories. The performance deteriorates as the factor $\omega^2 R$ gets away from five. It is appreciated in Table 9 that, as the radius of the circular trajectory is increased for a constant angular speed of $\omega = 1rad/s$, the Root Mean Squared Error (RMSE) is considerably increased. Again, in the calculation of the Error, the first eight seconds of the performance have been considered as setting time, which has been obviated. The order of magnitude of the Root Mean Squared Error is increased once the radius of the trajectory is greater than 6 meters, being $\omega^2 R > 5$

Radius $[m]$	$\Delta x[m]$	$\Delta z[m]$	$\Delta\dot{x}[m/s]$	$\Delta\dot{z}[m/s]$
4	0.1015	0.0663	0.1364	0.1530
5	0.1425	0.1100	0.1562	0.1600
6	0.6420	0.7742	0.6794	0.3237
7	2.3161	2.7381	2.0126	1.1205
8	4.3588	5.6204	2.9219	1.5570
9	7.5520	9.3508	3.9199	2.1060

Table 9: Root Mean Squared Error after settling time in horizontal and vertical position and speed with respect to a target UCM trajectory of $\omega = 1rad/s$ as a function of the radius of the circle, R.

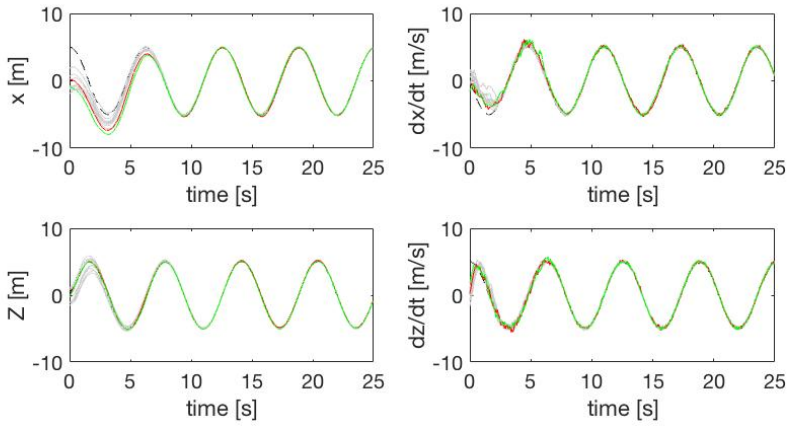


Figure 21: Uniform Circular Motion performance for $\omega = 1rad/s, R = 5m$. In green, the episode with the smallest position error; in red, the episode with the largest position error

In conclusion, the designed vehicle should have no problem performing any desired smooth trajectory (for instance, sinusoidal) if the given set of actions has been properly defined according to limiting conditions such as the ones that have been discussed in the case of a circle: maximum downward force and required projection of the thrust on the horizontal or vertical axes. Therefore, if the designer knows the trajectory that the MAV is going to follow, the definition of the set of actions is a previous and necessary task for the design of the vehicle.

4.5.2 Battlements

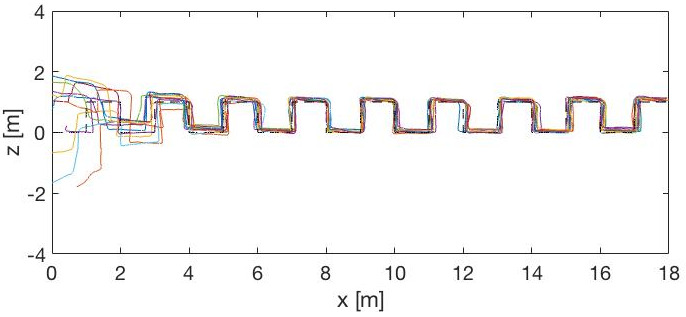
Imagine the case in which the vehicle trajectory includes straight flight and/or corners that have to be described with a considerable level of accuracy. The MAV should be capable of following any desired trajectory in order to be versatile. A priori, the fact that the only way for the proposed MAV to go down is to use zero thrust and fall may seem to restrain performance.

Figure 22 shows the performance of the flapping wing vehicle following a moving target that describes a series of battlements with constant velocity $V_G = 1m/s$. As it was predicted, according to the restricted set of actions, the vehicle never reaches a stable horizontal speed. It has to address its position at almost every time it takes an action. Figure 22a shows the trajectory of the vehicle since it can be difficult to figure it out by looking at the state variables evolution separately (Figure 22b).

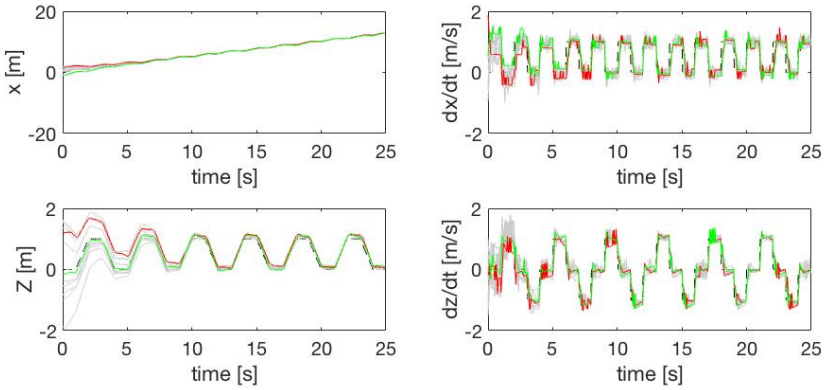
As one can imagine, the increase in speed will at some point deteriorate the performance of the vehicle since the gravitational force will not be enough to accelerate the vehicle to a specific velocity in the same period of time. For a battlements series of a four seconds period, it is shown on Table 10 that the Root Mean Squared Error of the vertical position significantly increases when the speed is

greater than $V = 3m/s$, a value that will be acceptable depending on the application for which the MAV is desired. This limit also depends on the duration of the period; the limiting velocity will increase for a greater period, since there is more time for the MAV to accelerate.

In conclusion, the designed autopilot makes it possible for the vehicle to perform smooth as well as rough sharped trajectories. However, the quality of the performance in both cases is limited according to the set of actions that has been assigned to the vehicle.



(a) Trajectory



(b) Evolution of state variables. In green, the episode with the smallest position error; in red, the episode with the largest position error

Figure 22: Performance of the MAV following a target trajectory describing battlements with $V_G = 1m/s$

Speed [m/s]	$\Delta x[m]$	$\Delta z[m]$	$\Delta \dot{x}[m/s]$	$\Delta \dot{z}[m/s]$
0.75	0.0849	0.0707	0.1857	0.1149
1	0.1010	0.0943	0.2640	0.2100
2	0.3122	0.3762	0.7750	0.6012
3	0.6324	0.7755	1.5302	1.1274
4	1.04302	2.8183	1.9666	2.0670
5	1.2657	7.1923	2.5119	2.7834

Table 10: Root Mean Squared Error after transition time in horizontal and vertical position and speed with respect to a target battlements trajectory of a four seconds period as a function of the speed.

4.5.3 Sudden changes in trajectory

The last challenge the already trained MAV will face is to follow a linear horizontal trajectory that suddenly changes its direction. In this way, the reaction capacity of the autopilot can be tested.

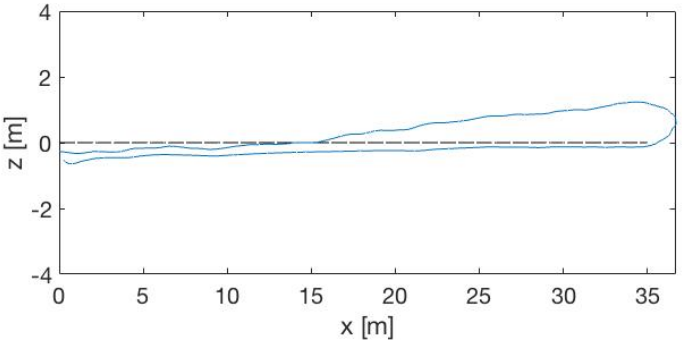
There is no special interest on simulating a sudden change in the vertical speed since the vertical motion performance of the vehicle is very simple according to the current set of actions. If the vehicle is going up and it suddenly needs to go down, it will apply zero thrust and reach the target negative velocity as soon as the acceleration of gravity allows it. On the other hand, if the MAV is falling down and it suddenly needs to go up, it will apply maximum thrust in the vertical direction until it reaches the target velocity. The imposed set of actions allows a pure vertical motion, so the reaction of the vehicle is predictable in these situations.

However, as it has been observed in previous sections, level flight is not an easy task for the MAV applying the current set of actions. The fact that the stroke plane angle can just be set at three different configurations limits the performance of the vehicle. Therefore, it seems interesting to check how far does this issue affect the performance of the MAV in case a sudden change of direction is required at horizontal level flight.

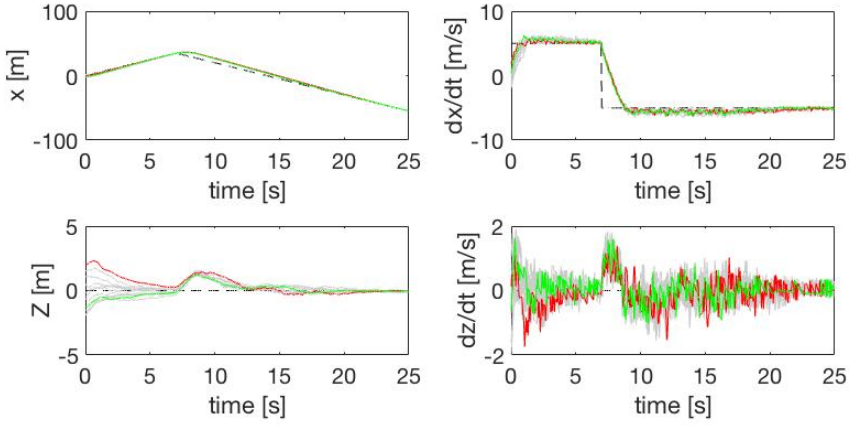
According to the results in Section 4.5.2, it would not be difficult for the MAV to react to a change in speed from $V_x = 1m/s$ to $V_z = -1m/s$, so that the test will be carried out for a more pronounced change that may present a challenge for it. Figure 23 shows the performance of the trained MAV when it encounters a step in velocity from $V_x = 5m/s$ to $V_x = -5m/s$. It is observed that, due to the reduced set of stroke plane angles, seeking for a larger projection of the thrust on the horizontal direction, the projection of thrust on the vertical direction has also been increased. This issue makes the vehicle get almost two meters away from the trajectory in the vertical position before reaching again the target trajectory. If the set of stroke plane angles was wider, the thrust projections would be easier to adjust and, therefore, the performance of the vehicle would be optimized in case it encounters this kind of

situation.

Moreover, despite the vehicle limitations, it is shown in Figure 23b that the vehicle is able to reach again the target trajectory, so that there is no delay between the target and the actual trajectory.



(a) Trajectory



(b) Evolution of state variables and actions. In green, the episode with the smallest position error; in red, the episode with the largest position error

Figure 23: Performance of the MAV when it encounters a step change in the target velocity

In conclusion, after all of the simulations that have been carried out with the designed autopilot, it seems that the performance of the vehicle could be considered acceptable for a real engineering application. The vehicle never crashes neither does it lose control if the set of actions has been defined according to the performance requirements.

4.6 Limits of a more realistic vehicle

A real physical device is not able to change of action instantaneously. Thrust may progressively adapt its module and its direction. Moreover, if the flow aerodynamic effects where entirely considered, the response to a new taken action would not become instantaneously effective. The problem becomes even more complex taking into account flapping wing aerodynamics.

In fact, according to the implemented algorithm, not only does the designed autopilot obviate the real behavior of the flow surrounding the vehicle and its effect, but it allows the module and direction of thrust to change instantaneously. Furthermore, the number of available actions is limited in order to produce an affordable size of the Q matrix, which leads to a big step size between the available thrust modules and stroke plane angles. These considerations seem to locate the designed autopilot far from reality, even more taking into account that no restraint has been imposed on the action that the vehicle can take.

Up to this point of the project, the autopilot was able to take whatever thrust/stroke plane angle combination that the policy map had established to be the best one at the given state the vehicle was found at. This means that if the vehicle was applying zero thrust at a specific moment, it could set maximum thrust instantaneously, although in a real case it should apply an intermediate value before reaching maximum thrust. In order to avoid this and make the simulation more realistic, the vehicle should be restrained to use only actions that are just a step further in the available set of thrust module and direction.

For this purpose, the same battlements trajectory test will be carried out again applying this restriction. Since it is the most difficult kind of trajectory for the MAV, the effects of restraining what action to take should be magnified in case it really affects performance.

Speed [m/s]	$\Delta x[m]$	$\Delta z[m]$	$\Delta \dot{x}[m/s]$	$\Delta \dot{z}[m/s]$
0.75	0.0700	0.0728	0.1817	0.1049
1	0.0917	0.1030	0.2598	0.1992
2	0.2907	0.3568	0.7790	0.5868
3	0.5909	0.7664	1,4811	1.1065
4	1.2887	2.9229	1.8505	2.0086
5	2.1249	5.7073	2.3238	2.7245

Table 11: Root Mean Squared Error after setting time in horizontal and vertical position and speed with respect to a target battlements trajectory of a four seconds period as a function of the speed for a restrained vehicle.

The results of the test can be checked on Table 11. Comparing these results to the ones obtained for the non-restrained vehicle on Table 10, it can be concluded that not only does the magnitude of the Root Mean Squared Error maintain unchanged,

but the average RMSE values are even slightly smaller for the restrained vehicle. It seems that the reduction of abrupt changes in thrust module and direction is slightly beneficial for the performance of the MAV, although the difference with respect to the initial autopilot is almost negligible.

According to the results, this correction on the initial MAV can be considered as satisfactory since it brings the vehicle closer to reality and it does not affect the quality of its performance. Therefore, this restriction will be applied on the following sections of the two degrees of freedom autopilot.

4.7 Circuit mission

The designed autopilot has been tested through the performance of periodic trajectories, but in this section, a possible engineering application has been defined for the vehicle to perform. The flapping wing MAV has been forced to follow a specific circuit of one minute duration trying to simulate a real mission, shown in Figure 24. The circuit is described counterclockwise. As it can be observed, the vehicle follows the predefined trajectory showing evidence of the shortcomings that have been discussed in Section 4.5, for instance, at the sinusoidal motion with the largest amplitude. In order to minimize the transition time, the MAV has been located at point $x = 0, z = 0$ at rest as initial condition.

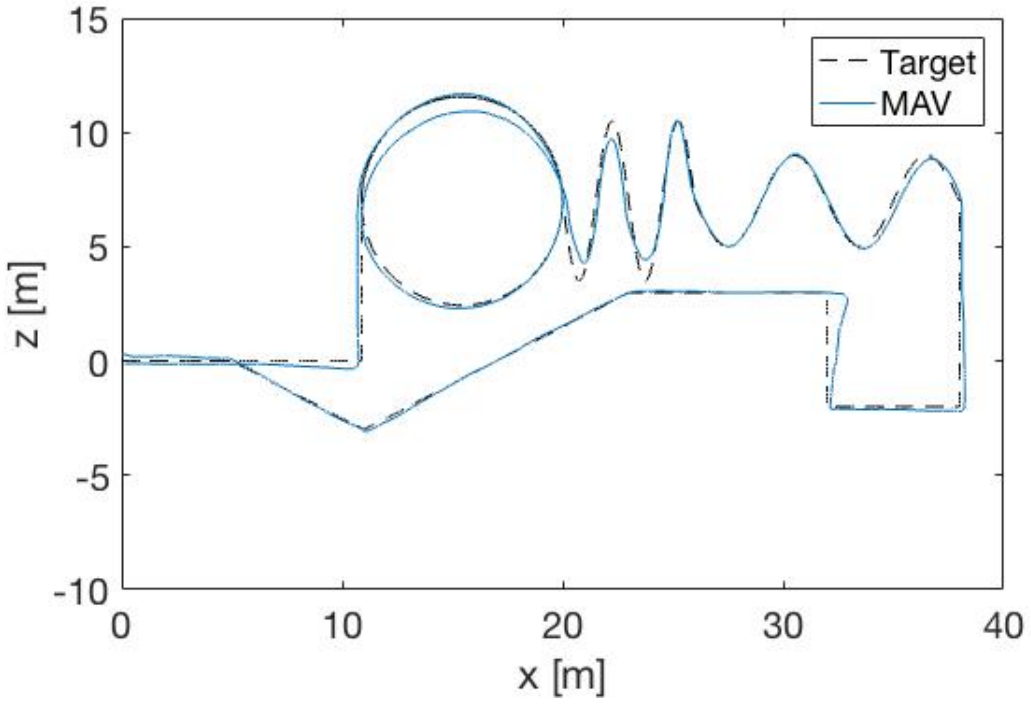


Figure 24: Trajectory of the vehicle describing the circuit mission

The evolution of the state variables of the MAV during the circuit can be observed on Figure 25. As it can be appreciated, the vehicle is able to successfully complete the circuit. It seems to find some difficulty to perform the second sinusoidal motion, but this does not mean that its evolution is not satisfactory since it does not diverge from the trajectory at any moment.

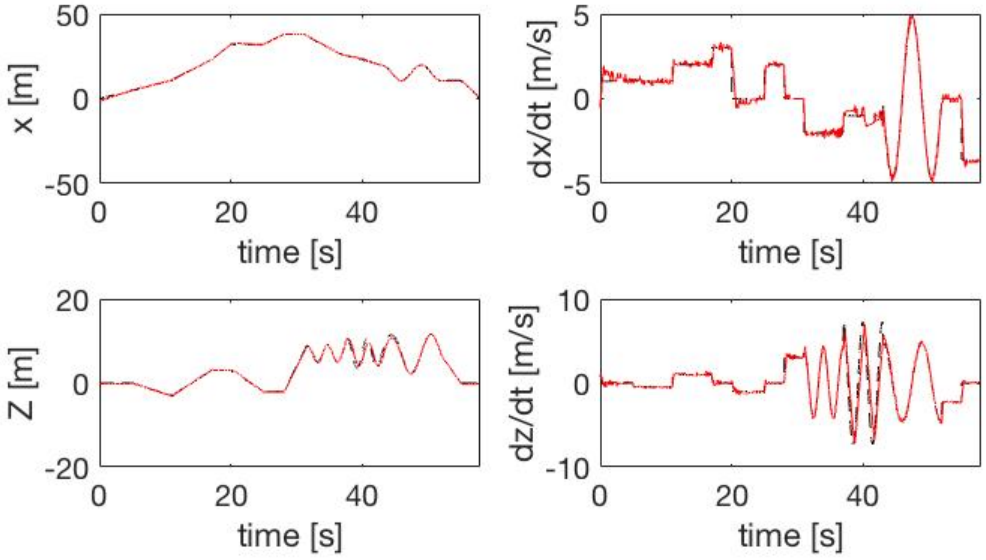


Figure 25: Evolution of the state variables and actions of the MAV performing the circuit trajectory

The vehicle has successfully completed the designed mission, so this example will be taken a step further. Energy consumption is one of the reasons for the development of an alternative MAV design apart from rotary wing. Taking into account the limited autonomy of quadcopters, the problem is exacerbated when the size of the drone is reduced, since a smaller battery implies a smaller capacity, leading to an even shorter flight duration. A point of interest from the engineering point of view is the energy consumption of the flapping wing vehicle. For this end, regarding the completed circuit, mission an attempt will be made in order to calculate the power consumption of the designed flapping wing vehicle.

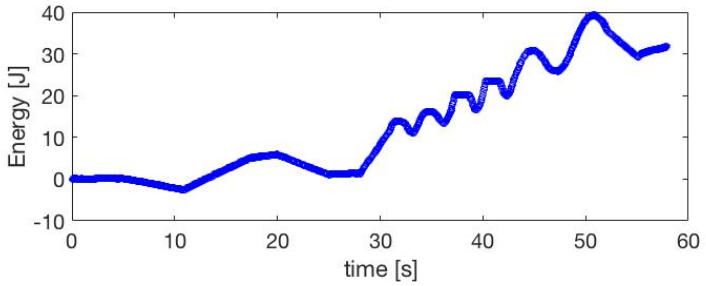
Now that the performance of the MAV has been considered to be satisfactory, it is time to calculate how much energy it has used. By definition, the energy that is developed due to the application of the thrust is equal to the time integral of the power developed by the thrust force

$$E = \int P dt \tag{26}$$

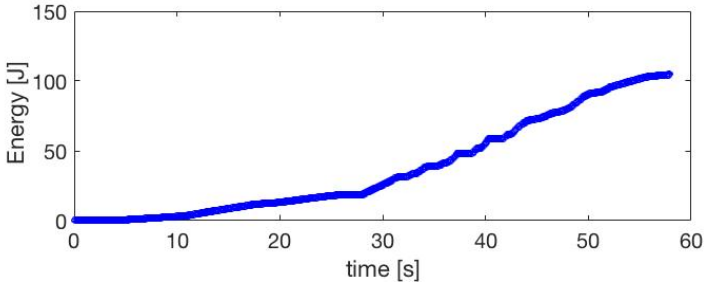
being the power evolution equal to the scalar product of the thrust force and the velocity of the moving body on which the force is applied

$$P = |T||V| \cos \psi \tag{27}$$

where ψ is the angle between the thrust force and the velocity vectors.



(a) Contribution of thrust to the kinetic energy of the system



(b) Evolution of the energy developed by the thrust force

Figure 26: Evolution of energy during the circuit mission

According to Equation 27, at those moments at which the ψ angle between the thrust force and the velocity of the MAV is greater than 90^{o}, power becomes negative and so does the increment of energy (Fig. 26a). This occurs, for instance, at the sinusoidal segments, where thrust may be opposite in direction to the velocity of the MAV in order to minimize tangential acceleration. However, one would expect that the consumption of energy is always increasing. This negative sign is just representing the contribution of the thrust force to the total energy of the system. In fact, if thrust and velocity are opposite in direction, the vehicle would be decelerating and in that case there is also consumption of energy.

Therefore, in order to estimate the amount of energy developed by the thrust force, the absolute value of the power developed by the thrust force has been calculated. Looking at Figure 26b, it can be appreciated that the total amount of developed energy is now increasing with time. At the end of the circuit, a total amount of 104.92 Joules has been developed by the thrust force. Considering that the circuit duration is 58 seconds, the average power developed by the MAV is $P_{developed} = 1.808W$, which is not equal to the average power required.

In order to develop a certain amount of power, a greater amount of power is required

because of the propulsive efficiency η_P

$$P_{Req} = \frac{P_{developed}}{\eta_P} \tag{28}$$

Any device experiences some loss of efficiency due to many reasons. For instance, in the case of the flapping wing vehicle, air friction and the mechanical system in charge of the flapping movement of the wing are responsible for the loss of efficiency. Therefore, since the aerodynamic model and the number of systems consuming energy in this problem are unknown, it is not possible to estimate what would be the average power required by the flapping wing vehicle. It would be possible to calculate the required power once the propulsive efficiency of the MAV is known.

4.8 Validity of the policy map

At this point of the study, the policy map has been designed and applied for a 100 grams vehicle. However, imagine that the same engineer wants to produce different size MAV's. Cost would be considerably increased if a new policy map had to be developed for every single MAV model. Therefore, the aim of this section is to check if the policy map that has once been developed is applicable to a vehicle that is different in size.

First, some considerations have to be made. It will be assumed that the wing loading of the vehicle keeps constant for all size models, $m/S = 6.89kg/m^2$, so that $S \propto m$. The drag of the vehicle will be considered as the one of a sphere ($C_D = 0.44$). Moreover, in the current 100 grams vehicle the maximum thrust has been set to be twice as large as weight, so that the MAV could perform a wide range of maneuvers. This relation will also be maintained, so that $T_{max} \propto m$. Regarding the stroke plane angle, the same set of 60, 90 and 120^{o} ($= 90 \pm 30^{\text{o}}$) will be kept for the moment since it corresponds to the original set of actions in the policy map.

Now that these design settings have been established, it is time to size the new vehicle on which the current policy map is going to be applied. Thinking of an engineering application in which, for instance, a camera is needed, a MAV of one kilogram is designed ($m_{new} = 1kg$), that is $m_{new}/m = 10$. In this way, $S_{new} = 10S = 0.145m^2$ and $T_{max,new} = 10T_{max} = 20N$.

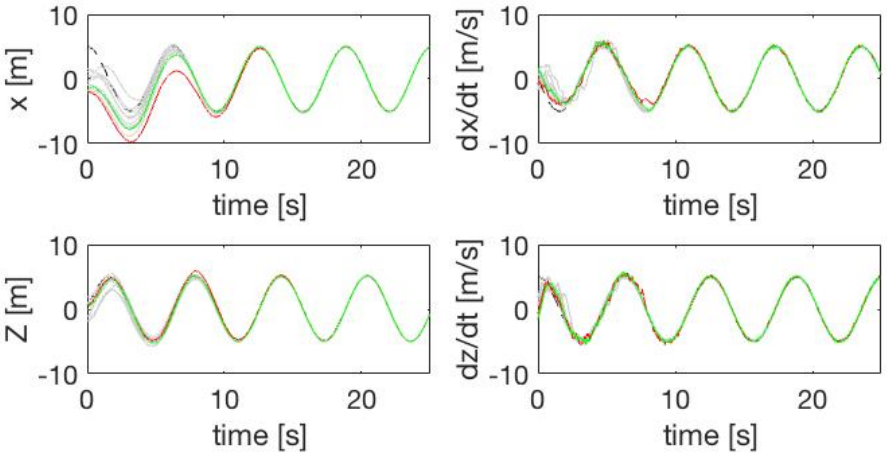


Figure 27: Uniform Circular Motion performance of a $1kg$ MAV using the policy map developed for the original vehicle, $\omega = 1rad/s, R = 5m$. In green, the episode with the smallest position error; in red, the episode with the largest position error

In order to test the policy map on the new vehicle, a uniform circular motion has been set to be the target trajectory. As it can be confirmed by looking at Figure 27, maintaining the vehicle properties (S, T_{max}) proportional to the mass, the policy map that is created for an initial case can be used for other sizing cases obtaining the same performance quality. The setting time for the new vehicle is very similar to the one shown on Figure 21 for the initial vehicle, around eight seconds. This property of the policy map reduces the cost of the development of the preliminary autopilot since it just needs to be developed once. Take into account that for this test the wing loading has been maintained constant and also the stroke plane angle range.

If this study is further followed, it would be also beneficial for the reduction of the policy map cost to check if it is valid for a different set of stroke plane angles. Since the original policy map was created for a set of nine actions (3 thrust modules x 3 possible β), the number of possible stroke plane angles has to be maintained to three. Therefore, the maximum and minimum stroke plane angles can be changed in order to check its effect on the performance of the policy map. For this simulation, the new vehicle of one kilogram will be used.

Thinking of a real flapping wing vehicle, let's consider that the range of the swept stroke plane angle is not greater than ninety degrees. Then, keeping the middle value at 90^{o}, the new vehicle will be able to flap its wings at $\beta = 90 \pm 45^{\text{o}}$, so that the possible stroke plane angles are 45, 90 and 135^{o}. Going a step further, according to Equation 25, for the optimum action that the MAV can perform at point B of the UCM (Fig. 18) with the actual controls $(T = 10N, \beta = 45^{\text{o}})$ the performance limit at that point has changed, so that

$$\omega^2 R \leq 7.07 \tag{29}$$

This means that the new vehicle can perform a wider range of uniform circular motion trajectories with great accuracy. Furthermore, as Figure 28 and Table 12 show, the new vehicle with the new set of actions is capable of following a circular target trajectory of $\omega^2 R = 7$ with the same order of magnitude RMSE that the original MAV when it was found at the limiting condition of $\omega^2 R = 5$. In the same way, as the $\omega^2 R$ factor gets away from the limit condition, the order of magnitude of the Root Mean Squared Error considerably increases.

Radius $[m]$	$\Delta x[m]$	$\Delta z[m]$	$\Delta \dot{x}[m/s]$	$\Delta \dot{z}[m/s]$
6	0.1375	0.2086	0.2254	0.1634
7	0.2302	0.2285	0.8526	0.2232
8	1.0175	0.9579	0.9987	0.9080
9	3.1100	2.9212	2.5829	2.4266

Table 12: Performance of the MAV of 1kg and $\beta = (45, 90, 135)$ degrees using the policy map for the original vehicle. Root Mean Squared Error after setting time in horizontal and vertical position and speed with respect to a target UCM trajectory of $\omega = 1 rad/s$ as a function of the radius of the circle, R.

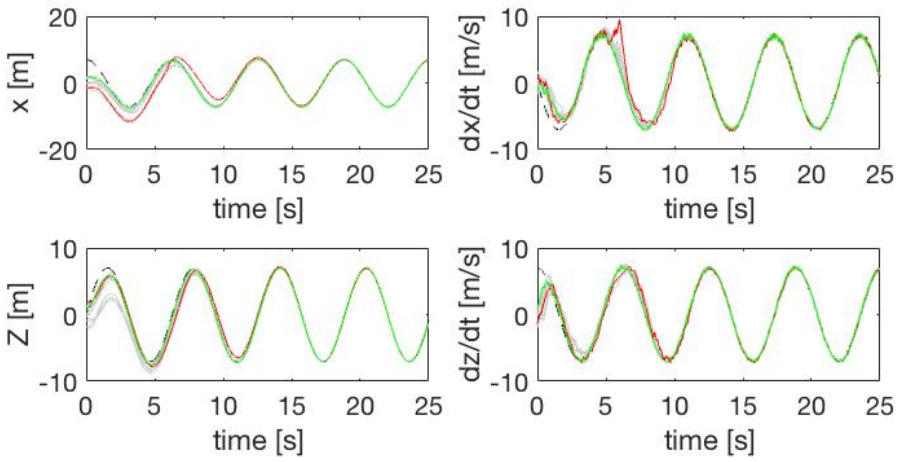


Figure 28: Uniform Circular Motion performance for a $1kg$ MAV with $\beta = (45, 90, 135)$ degrees using the policy map developed for the original vehicle, $\omega = 1 rad/s, R = 7m$. In green, the episode with the smallest position error; in red, the episode with the largest position error

After this analysis, it can be concluded that the cost of the production of the policy map gets drastically reduced. Not only is the same policy map valid for MAV's of different weight, but it is also applicable to other MAV's that, following the same

definition of the set of actions, have different stroke plane angle values. Thanks to this study, once the designer has made their mind clear about the wing loading of the vehicle and the definition of the set of actions, the policy map of the vehicle just needs to be produced once. If a change wants to be made that does not affect to the wing loading or the definition of the stroke plane angle set, the original policy map is equally valid for the new vehicle, saving time and efforts.

4.9 Reduction of the state space

As it was appreciated in Section 4.2, the number of rows of the Q matrix grows exponentially with the number of state variables. This issue can become a disadvantage regarding the future design of a complete longitudinal control autopilot, where the pitch angle and rate have to be introduced. The size of the Q matrix and the resolution of the state space has to be such that the learning process is not extremely long and that the CPU can afford the size of the matrix. Moreover, the resolution of the state space has to be such that it describes the state space in a way that the policy map does not miss any strategic point of space where there should be changing actions to take.

Up to this point of the two dimensional autopilot study, it has been shown that the reduction in resolution with respect to the one dimensional case has not affected the performance quality of the autopilot. Instead, it has produced a reduction in the size of the policy map without missing accuracy at the moment of assigning actions to points in the state space. This resolution of the state space can therefore be considered to provide a faithful description of space. If this resolution of space is maintained, something has to be done in order to reduce the size of the policy map further without deteriorating the performance of the MAV.

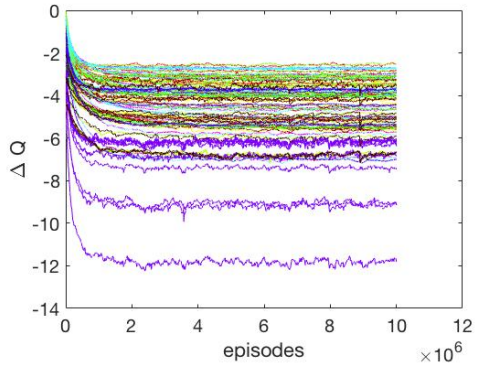
The policy map of the two degrees of freedom autopilot is not straight forward to represent. This was one of the reasons for the analysis of the one dimensional autopilot. If one looks at the policy maps on Figure 12, it seems that making zoom on the policy maps one would still have almost the same policy map. This means that there is part of the policy map that could be obviated. This leads to the conclusion that maybe, if the range of the state space was reduced for the current resolution, the reduced policy map would still be valid for the MAV without depraving its performance.

The aim of this section is to evaluate the effect of the range reduction of the two-dimensional problem state variables. For that purpose, the original 100 grams vehicle will be used. If the reduction of the state space does not significantly affect the training and performance of the MAV, it can be concluded that a valid policy map can be produced with a very reduced Q matrix. With regard to the future development of a complete longitudinal controller, involving three dimensions (x, z, θ), a size reduction of the two d.o.f. problem Q matrix would be very useful in order to make the size of the three d.o.f. problem Q matrix affordable. For this end,

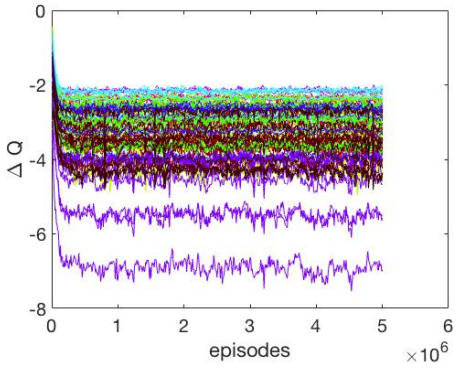
three different reductions have been selected, as it is shown on table 13, together with the size of their respective Q matrices. As a reminder, the resolution that was defined in Section 4.2 has been maintained.

Case	Q matrix rows	x Range	z Range	$\dot{x}$ Range	$\dot{z}$ Range
0	810000	[-2,2]	[-2,2]	[-2,2]	[-2,2]
1	16384	[-0.5,0.5]	[-0.5,0.5]	[-1,1]	[-1,1]
2	4096	[-0.5,0.5]	[-0.5,0.5]	[-0.5,0.5]	[-0.5,0.5]
3	1024	[-0.25,0.25]	[-0.25,0.25]	[-0.5,0.5]	[-0.5,0.5]

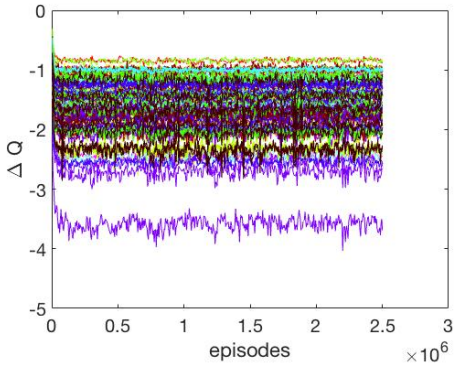
Table 13: Proposed ranges for the reduced training state space of the MAV



(a) Case 1



(b) Case 2



(c) Case 3

Figure 29: Evolution of the values of the Q matrix for Cases 1, 2 and 3

As it was stated in Section 4.2, the original range of the state space resulted in a Q matrix composed of 810000 rows and 9 columns (Case 0). Since the same set of actions is going to be used to test the new proposed ranges, the number of columns is maintained. As it is shown on Table 13, the number of rows of the Q matrix has been substantially reduced. Appreciate that Case 1, which is the proposal with the largest number of rows of the Q matrix, has reduced this number in a 98%

with respect to the original state space ranges. With this reduced state space, the learning process becomes leaner.

In the same way that has been done for every learning process, the evolution of the mean values of the Q matrix is shown on Figure 29 for the three proposed cases on Table 13 representing the same zones that in Section 4.3. It is appreciated on the Figure that, as the number of elements of the policy map is reduced, the absolute value of convergence for the given colour zones decreases since the policy map converges in a smaller number of episodes. The oscillation of the mean values through the learning process does not mean that the given policy maps will be less effective, as it was shown in Section 3.6 for the resolution study.

Since all the policy maps have converged, their performance can be tested. For this end, the trained MAV will be forced to follow again a Uniform Circular motion of radii $R = 1m$ and angular speed $\omega = 1rad/s$ since it was checked that the original MAV was able to successfully perform this type of trajectory.

Case	$\Delta x[m]$	$\Delta z[m]$	$\Delta \dot{x}[m/s]$	$\Delta \dot{z}[m/s]$
0	0.0374	0.0316	0.1200	0.1241
1	0.0854	0.0361	0.1245	0.1122
2	0.0548	0.0374	0.1204	0.1296
3	0.0374	0.0346	0.1396	0.1229

Table 14: Root Mean Squared Error after transition time in horizontal and vertical position and speed with respect to a target UCM trajectory of $\omega = 1rad/s$ and $R = 1m$ for the different range proposals.

The results on Table 14 show that, for all the different range reduction proposals, the Root Mean Squared Error with respect to the target trajectory keeps at the same order of magnitude that in the original case. In fact, according to the actual resolution, in order to take an action, the MAV just needs to know the sign of the state variables with respect to the moving target reference frame. Therefore, the smallest design of the policy map would be equally valid compared to the original policy map. After these results, the production of the policy map becomes even more economical; time and memory space is saved, as well as efforts.

These conclusions will be considered in further sections. The main advantage of this reduction of the state space is that it significantly simplifies the definition of the complete longitudinal autopilot, since it makes possible the production of an affordable Q matrix.

5 Three Degrees of Freedom Problem

The aim of this chapter is to define the preliminary approach of a complete longitudinal control autopilot for a simple model of flapping wing vehicle. Up to this point of the project, the MAV has been treated as a point particle that is just defined by its x and z coordinates in the vertical plane. However, real Micro Air Vehicles should be treated, at least, as rigid bodies. Therefore, although the two degrees of freedom problem looked like a close approach to the desired autopilot, it does not account for the real motion of a flapping wing vehicle.

Now that the motion of the MAV in the vertical plane as a point particle has been completely controlled, the development of the preliminary design of the autopilot can be taken a step further introducing pitch control. The introduction of rotation into the dynamic model will make the simulated MAV be closer to a real vehicle. In fact, with the introduction of the pitch angle, the MAV will now be treated as a rigid body, including translation and rotation.

For this end, all the conclusions that have been derived from previous sections will be considered. Since the introduction of a third state variable increases the computational cost of the problem, the development of the one and two d.o.f. autopilots will be helpful in order to define and tackle this new problem.

Due to the large computational cost that this study requires, the vehicle will be trained to reach hover flight. Hover and Uniform Circular Motion will be the missions it will have to perform through the application of the exploitation algorithm. It would be possible to train the vehicle to perform infinite maneuvers in this three d.o.f. problem with a more powerful CPU. The validity and the potential of the Q-learning algorithm has been thoroughly demonstrated for the one and two degrees of freedom problems.

5.1 Problem definition

As it has been introduced, in order to define a problem that is closer to the real motion of a MAV, a new state variable will be introduced in this part of the project: the pitch angle θ. In fact, thanks to the introduction of the pitch angle, the thrust vector would be able to adopt unlimited directions in this problem. For instance, the limitations that the vehicle showed in the vertical direction in the two d.o.f. problem could be overcome. This would make the MAV more agile, but it will also make the problem much more complex, for instance, making the exploration process much longer.

The three degrees of freedom of the vehicle in this new case study are x, z and θ. The dynamic model of this problem is shown on Figure 30. As it can be appreciated, this three d.o.f. problem includes two new geometric parameters that define the vehicle: l, which is the distance from the center of gravity G to the center of rotation of

the stroke plane (C) and L, the length of the vehicle. There are three forces that are responsible for the motion of the vehicle. Thrust is applied at point C at a distance l from G, the aerodynamic drag is applied at the leading edge at a distance $L/2$ from G and the weight of the vehicle is applied at the center of mass G. An aerodynamic moment $M_{y,LE}$ is applied at the leading edge of the vehicle according to the slender body theory (Chapter 8.3. of [12]). This theory establishes that the air flow surrounding a slender body creates on it a pressure distribution that results in an external aerodynamic pitching moment that has to be considered in the equations of motion. For the application of the slender body theory, the body of the flapping wing vehicle will be considered as a slender bar of length L and radii R.

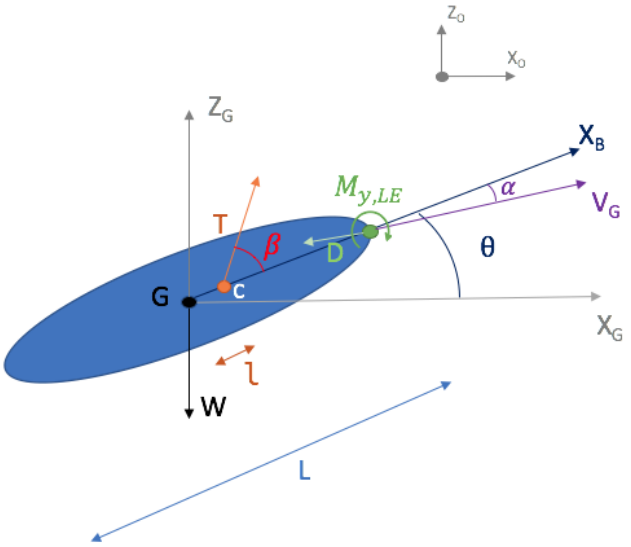


Figure 30: Dynamic Model for a three dimensional motion

According to the slender body theory, the aerodynamic moment at the leading edge would follow the next expression

$$M_{y,LE} = +\alpha\rho U_\infty^2 (S(L)L - V) \tag{30}$$

where α is the angle of attack of the body, U_∞ is the aerodynamic velocity of the vehicle, V is the volume of the body and $S(L)$ is the cross-sectional area of the vehicle at the trailing edge. This expression can be rewritten as

$$M_{y,LE} = +\alpha\rho U_\infty^2 V(S(L)L/V - 1) = +\alpha\rho U_\infty^2 V C_m \tag{31}$$

where $C_m = S(L)L/V - 1$, becoming a geometry dependent parameter. As it would be expected, at an angle of attack of 180^{o}, the aerodynamic moment should be equal

to zero due to axial symmetry. Therefore, the final expression for M_y, LE has been modified and it will be applied as

$$M_{y,LE} = +\sin\alpha\rho U_\infty^2 V C_m \tag{32}$$

Since no wind is considered in the simulations, the aerodynamic velocity is equal to the velocity of the vehicle ($U_\infty = V_G$). Knowing this, and having a look at Figure 30, the equations of motion for the three degrees of freedom problem can be derived through the application of second Newton's Law and the conservation of angular momentum about the y_B axis

$$m\frac{d\dot{x_G}}{dt} = T\cos(\beta+\theta) - \frac{1}{2}\rho S V_G^2 C_D(\frac{\dot{x_G}}{|V_G|}) \tag{33}$$

$$m\frac{d\dot{z_G}}{dt} = T\sin(\beta+\theta) - mg - \frac{1}{2}\rho S V_G^2 C_D(\frac{\dot{z_G}}{|V_G|}) \tag{34}$$

$$I\frac{d\dot{\theta}}{dt} = M_{y,LE} + \frac{1}{2}\rho S V_G^2 C_D(\frac{\dot{z_G}}{|V_G|})\frac{L}{2}\cos\theta - \frac{1}{2}\rho S V_G^2 C_D(\frac{\dot{x_G}}{|V_G|})\frac{L}{2}\sin\theta - Tl\sin\beta \tag{35}$$

where I is the inertia of a slender bar of mass m and length L, $I = \frac{1}{12}mL^2$.

The definition of the set of actions in this case is very similar to the one in the two d.o.f. problem. The only difference is that the stroke plane angle β is now defined as the angle between the longitudinal body axis x_B and the normal vector of the stroke plane, coincident again with the direction of thrust.

Once the equations of motion have been written, the reward function has to be defined so that the vehicle can have a quantitative measure to differentiate what is 'good' from what is 'wrong'. In the three d.o.f. problem, only the pitch rate is going to be included in the reward function since the pitch angle is going to be determined for a given flight condition in the equations of motion.

$$\begin{aligned} R(t) = -\phi(|x_G(t) - x_0(t)|^2 + |z_G(t) - z_0(t)|^2) \\ -(1-\phi)(|\dot{x_G}(t) - \dot{x_0}(t)|^2 + |\dot{z_G}(t) - \dot{z_0}(t)|^2) - \lambda|\dot{\theta}(t) - \dot{\theta_0}(t)|^2 \end{aligned} \tag{36}$$

The goal in the exploration algorithm is to reach equilibrium, hover, so that the success condition now includes $\dot{\theta} = 0$. The parameter that is going to establish the weight of the error in the pitch rate is λ, which can be observed in the new reward function (Eq. 36). The training condition for the new problem is $x_0(t) = 0, z_0(t) = 0, \dot{x_0}(t) = 0, \dot{z_0}(t) = 0, \dot{\theta_0}(t) = 0$, which, according to the equations of motion, implies $\theta = 90^{\text{o}}$.

The value of ϕ will be set in this problem according to the conclusions derived in Section 3.5. It is convenient to study what is the effect of the new parameter λ on the exploration phase in this new problem.

5.2 Definition of the vehicle and conditions

Now that the dynamics of the three d.o.f. problem have been defined, the new vehicle can be designed. The design of this vehicle is going to be based on the design criteria of the two d.o.f. problem as far as it is possible. Remember the LIPCA-actuated flapping wing device.

Basing the design of the new vehicle on the criteria in Section 4.2, the properties that will be maintained are

- $m/S = 6.897 kg/m^2$
- $C_D = 0.44$
- Frequency of taking an action = 10 Hz

According to Equation 35, the vehicle will experience a lower angular acceleration the larger the inertia of the body about the y_B axis. Therefore, in an attempt to reduce the angular acceleration, the new vehicle will have the following properties

- $m = 1kg$
- $S = 0.145m^2$

That is all the design criteria that can be inherited from previous sections. As it has been said, the three d.o.f. problem involves new design parameters, such as the length of the vehicle L, the point of application of the thrust (given by l) and Cm, related to the volume distribution of the vehicle. Moreover, in order to simplify the calculation of the inertia of the body about the y_B axis, the vehicle will be considered as a slender bar of mean radii R.

Some of these parameters are going to be fixed thinking of a reasonable geometry or size for a one kilogram vehicle. For instance, the length and radii of the vehicle will be set as

- $L = 0.3m$
- $R = 0.06m$

There are still two properties pending to be defined: l and C_m. In order to determine these two properties, a short study has been carried out. In this study, cruise conditions have been set for a velocity of $V_G = 7m/s$, at equilibrium according to the equations of motion and fixing all the previous vehicle characteristics. The study has been carried out at Sea Level conditions, so that $\rho = 1.225 kg/m^3$ and $g = 9.81 m/s^2$. Different combinations of C_m and l have been introduced in the

equations in order to obtain the lowest possible pitch angle and a value of the stroke plane angle that is inside the available range. Note that the set of actions has still not been fixed, but a stroke plane angle that is not greater than thirty degrees should be reasonable. Finally, after the study, a choice has been made on fixing

- $C_m = 2$
- $l = \frac{1}{8}L = 0.125m$

since these values allow cruise at a pitch angle of $\theta = 62.45^{\text{o}}$ applying a stroke plane angle of $\beta = 16.50^{\text{o}}$.

Regarding the set of actions, it will be defined in the same way as in Section 4.2, presenting three possible values for the module of thrust (maximum thrust, half maximum thrust and zero thrust) and three different stroke plane angles (-30º,0 and 30º). Therefore, the maximum thrust will be considered again as $T_{max} \approx 2mg$, so that

- $T_{max} = 20N$

The only settings that still have to be defined are the resolution and the ranges of the state space. Regarding resolution, since the two d.o.f. problem satisfactorily worked with the assigned resolution, it is going to be almost maintained for the discretization of $x, z, \dot{x}$ and $\dot{z}$. Nonetheless, since the introduction of the pitch angle and pitch rate is going to considerably increase the size of the Q matrix, the ranges of these variables are going to be reduced. It has been shown in Section 4.9 that the reduction of the state space of these variables does not deteriorate the performance of neither the exploration and exploitation algorithms. For redundancy, the ranges applied to the Case 3 in Section 4.9 will be applied, so that

- $-0.5 \leq x \leq 0.5m$
- $-0.5 \leq z \leq 0.5m$
- $-0.5 \leq \dot{x} \leq 0.5m/s$
- $-0.5 \leq \dot{z} \leq 0.5m/s$

It is time to discretize the two state variables that have been introduced in this problem: the pitch angle and the pitch rate. The range of the pitch angle will be such that it covers all the possible values that it can adopt

- $0 \leq \theta < 2\pi \ rad$

Moreover, the range for the pitch rate has been set to be

- $-\pi \leq \dot{\theta} \leq \pi \ rad/s$

The number of elements for the discretization of $x, z, \dot{x}$ and $\dot{z}$ will be 9 each. Since there might be important changes in what action to take regarding the pitch angle at which the vehicle is found, the pitch angle will be discretized in 12 elements. Since the angular speed may show a fast variation, it will be discretized into 7 elements.

This combination makes the number of rows of the Q matrix be equal to 551124. This number is large taking into account that the applied discretization may not lead to a very effective learning process. Moreover, the rough resolution may lead to a longer exploration phase if the state space does not represent reality as thoroughly as it should be required.

Again, the size of a time-step in the integration of the dynamics will be equal to $h = 0.05s$.

5.3 Effect of the parameters of the Reward Function

The aim of this section is to study the effect of the λ parameter in order to optimize the exploration process of the three d.o.f. problem.

As it was said in the definition of the three degrees of freedom problem, there is a new parameter having a role in the reward function: λ. This parameter accounts for the weight of the quadratic error of the pitch rate. One would expect that a low λ would make the exploration algorithm show a low concern about making the vehicle reach zero pitch rate. On the other hand, having a large λ would lead the training algorithm to consider as a success any state at which the pitch rate is very close to zero, even if the vehicle is far from the desired state looking at the other state variables.

5.3.1 Learning process

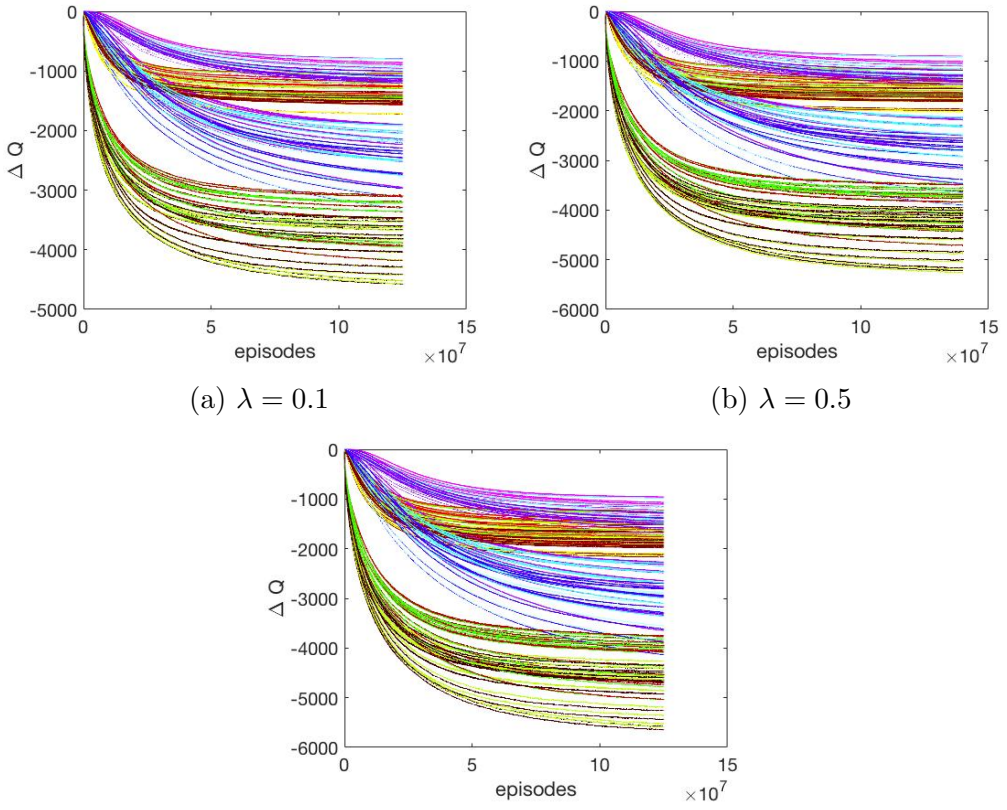
In this problem, the results of the parametric study carried out in the one d.o.f. problem will be considered again, so that

- $\mu = 0.1$
- $\gamma = 0.9$
- $\phi = 0.5$

Since a new parameter has been included in the reward function, it is important to carry a new study in which its effect is evaluated. In this way, the performance of the learning process of the three degrees of freedom problem can be optimized. A set of three different values of λ has been selected: $\lambda = 0.1, \lambda = 0.5$ and $\lambda = 0.9$. These values will be enough in order to have an approximation of what is the best parametric combination for this problem.

In order to evaluate the evolution of the exploration process, the same four zones defined in section 4.3 will be studied applying the same colour criteria (Tables 6, 7 and 8). The evolution of the mean values of the Q matrix in these sixteen zones has been represented in Figure 31. It is appreciated on the Figure that the mean absolute values for the are far larger than the values found in the previous studies. This is

due to the complexity of the three d.o.f. problem, since the learning algorithm has to apply forceful corrections in order to train the vehicle. Furthermore, the distribution of the colour lines is very similar for the three different values of λ, meaning that the weight of the pitch rate in the reward function does not strongly affect the performance of the learning autopilot.



(a) $\lambda = 0.1$

(b) $\lambda = 0.5$

(c) $\lambda = 0.9$

Figure 31: Evolution of the values of the Q matrix for changing λ

Since the three degrees of freedom problem shows very high computational costs, the exploration algorithm has been stopped in this case before total convergence. The colour lines have not become horizontal yet, but the learning that has been carried out until that moment should be sufficiently satisfactory. As a reminder, in Figure 31 there are nine lines of the same colour, each one representing the expected reward of taking a specific action at the zone represented with that colour. The colour line (out of the total nine with the same colour) that converges at the highest value corresponds to the action that the policy map has found out to be the best choice at that zone. Therefore, although the policy map is not completely converged, it is considered to be sufficiently 'good' while the upper lines of the colour zones do not seem to cross any of the lines representing the same zone, meaning that the best action at that zone will not be changing.

5.3.2 Performance

The difference in the evolution of the mean values of the Q matrix is very small for changing λ. No clear conclusions can be made by looking at Figure 31. Therefore, the execution of the exploitation algorithm will be essential in order to evaluate what is the optimum value for the λ parameter in order to achieve the best hover performance possible.

Figure 32 shows the performance of the vehicle trying to reach hover conditions. Again, the sample episode showing the minimum mean position error is represented in each case with green lines and the episode of maximum position error is represented in red. Having a look at the Figure it seems that $\lambda = 0.9$ will be the best option since the number of times the vehicle falls is the smallest for the selected period of time. In fact, the three values of λ have allowed at least one episode that has not diverged from hover, which means that the learning process was the proper one.

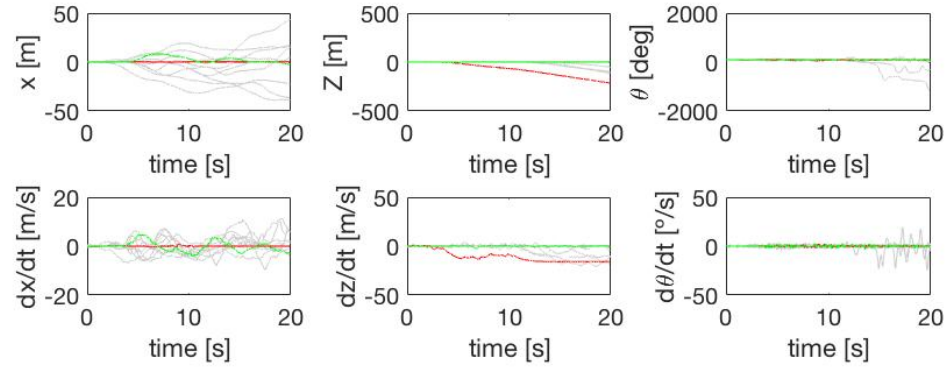
In order to make sure that this conclusion is true, the quality of the performance will be measured according to the average percentage of crashes that each λ value produces for a total number of 1000 episodes. It has been considered that a crash takes place when the position RMSE is greater than 15 meters. Looking at Table 15, it can be confirmed that the value that minimizes the number of crashes of the vehicle is $\lambda = 0.9$, showing a lower percentage of crashes than the obtained for the other two λ values. Therefore, in order to train the vehicle to perform complete longitudinal control in hover conditions the greatest learning quality is reached when the weight of the pitch rate in the reward function is greater than the weight of the rest of the state variables involved.

λ	Crashes
0.1	99.6%
0.5	98.5%
0.9	90.4%

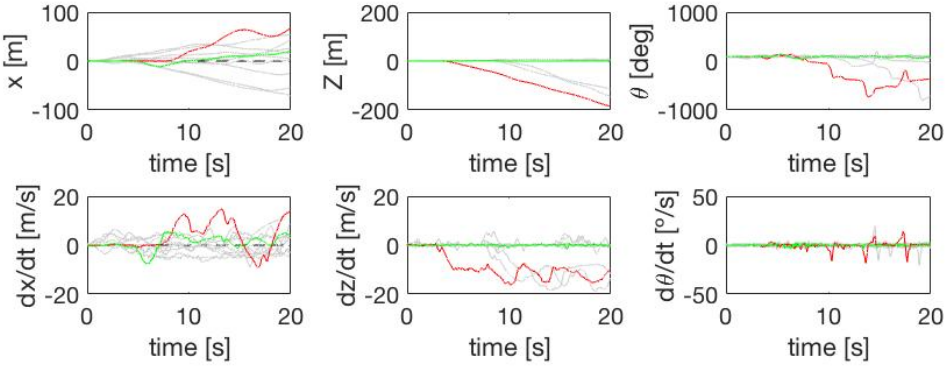
Table 15: Percentages of the number of crashes as a function of the λ parameter. The vehicle crashes when the mean position error in the episode is greater than 15 meters. Total number of episodes:1000

However, Table 15 shows that the accuracy of this three degrees of freedom autopilot is far from the one of the one and two d.o.f. problems. There are several possible reasons for this. The first one is that the learning process has been stopped before it had really finished, so that maybe the policy maps that have been used in the exploitation algorithms differ from the totally converged policy maps. The time cost of the production of the three d.o.f. policy map has been a limitation for the development of this last problem. The second possible reason is that, although the reduction of the state space range in the two d.o.f. did not affect the learning

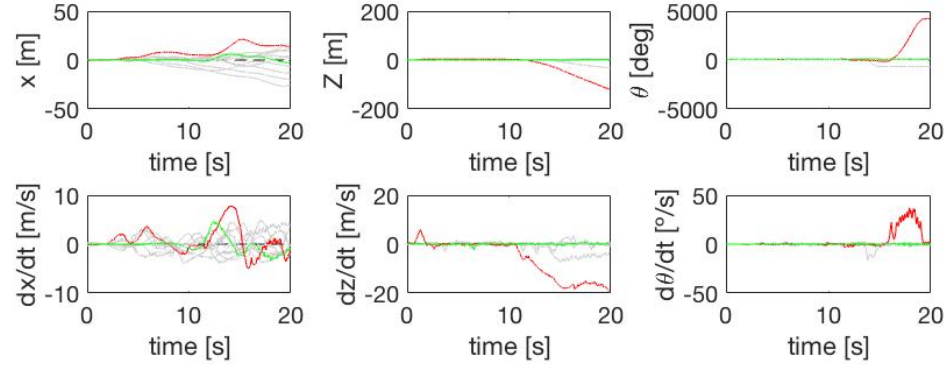
performance of the MAV, maybe a larger state space in the three d.o.f. problem will produce a better policy map. Therefore, another restrain that has been present in this problem is the size of the Q matrix because of the finite CPU RAM Memory.



(a) $\lambda = 0.1$



(b) $\lambda = 0.5$



(c) $\lambda = 0.9$

Figure 32: Hover performance for changing λ. In green, the episode with the smallest position error; in red, the episode with the largest position error

In the same way that has been done in the one and two d.o.f. problems, once the vehicle has been trained to reach hover conditions at a specific point, that policy

map should also be used to make the vehicle reach a moving target. As it is shown in Figure 33, with the policy map obtained for $\lambda = 0.9$, the vehicle tries to follow a Uniform Circular Motion, but most of the times it looses control because the quality of the current policy map is not enough. A successful trajectory would be reached with a more accurate resolution and range of the training state space.

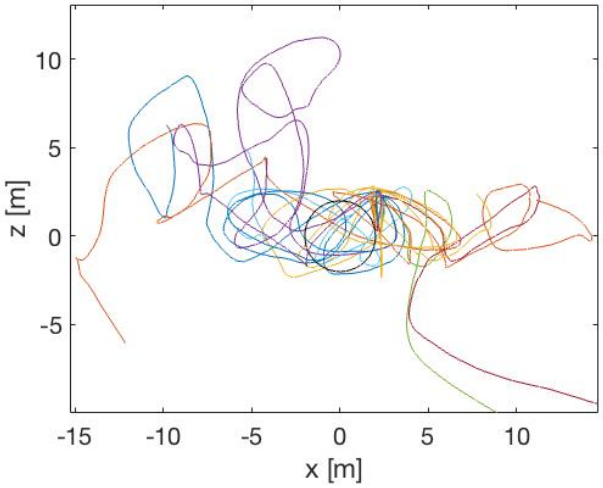


Figure 33: Trajectories described by the MAV trying to follow a UCM of $\omega = 1 rad/s$ and $R = 2m$ ($\lambda = 0.9$). In black, the target trajectory. Ten random episodes have been plotted using different colors

6 Conclusions

Throughout this project, a reinforcement learning algorithm has been developed in order to create a preliminary design of the autopilot of an autonomous MAV whose aerodynamic model is almost an unknown. This autopilot has been applied to the case of a flapping wing MAV, since the control of non-linear flight dynamics in this type of vehicle is nowadays a challenge because there is very little information about its aerodynamic model.

The main objective of the project was reaching longitudinal control of the vehicle. In order to reach an autopilot that accounts for the three degrees of freedom of longitudinal control in a rigid body, a preliminary one degree of freedom problem was elaborated, allowing a gradual introduction of degrees of freedom. This gradual development of the designed autopilot has allowed a deep understanding of the Q-learning algorithm that has been implemented and its optimization.

The split of the problem into the isolated exploration and exploitation phases has allowed to mark the difference between training and performance. This has been useful in order to concentrate efforts in the training process while having an overview of the evolution of the policy map and, after that, checking the quality of the performance of the produced policy map. Furthermore, in the case of the two and three d.o.f. problems, it has allowed the execution of the exploration algorithm into a Work Station and the performance of the exploitation algorithm into a Personal Computer separately, accounting for time efficiency.

It has been shown that the proper combination of the parameters that are involved in the Q-learning algorithm is crucial for the quality of the vehicle training process. Furthermore, the awareness of the effect of resolution and the size of the training state space make it possible to minimize the computational costs and optimize at the same time the efficiency of the learning process. The reduction of the size of the state space and the implementation of a more rudimentary resolution have not deteriorated the effectiveness of the training process.

Moreover, it has been demonstrated that a policy map that has been produced for a specific vehicle design is equally valid and efficient for the case of a vehicle of different dimensional characteristics and set of actions. The only two things that have to be maintained are the wing loading and the definition of the set of actions, so that the developed Q matrix is applied at an equivalent problem. This property leads to a remarkable reduction of the production cost of a policy map.

Computational costs have become a challenge in the development of the three degrees of freedom problem. That is why the development of the one and two degrees of freedom problems have been essential in order to reach all the above mentioned conclusions, simplifying the definition of the complete longitudinal control problem. It has been concluded that the reinforcement learning algorithm that has been implemented throughout the project could be useful for the preliminary design of an

autopilot that accounts for complete longitudinal control since, despite the policy map limitations, hover conditions have been reached.

6.1 Future works

As it has been said before, the way to tackle the proposed problem has been through the isolation of the exploration and exploitation phases. However, this procedure could be improved, for instance, if the exploitation algorithm was also used to train the vehicle. In this way, the exploitation algorithm would have the additional task of correcting any possible mistake that has not been perceived in the exploration phase. Therefore, the training process would not be finite, so probably it would not be necessary to keep running the algorithm until convergence, but it could be stopped sooner. Moreover, the actual training process that has been used in the project could be modified so that the probability of exploring by taking a random action was increased or reduced according to the designer preferences.

In addition, although in the three degrees of freedom problem the vehicle was able to reach hover conditions, the results could be improved if a more accurate resolution or a larger state space were applied. These changes would lead to a considerable increase of the size of the Q matrix that, using the actual equipment, would extend the time spent on the learning process up to several weeks, or even months. Actually, the policy maps that have been produced in the three degrees of freedom problem have been obtained after the execution of the exploration algorithm for more than 72 hours.

In order to reduce the time required for the exploration phase of a bigger in size Q matrix there are two different suggestions. The first one and the simplest is to use a more powerful computer with a greater amount of processors, for instance. The second one is to explore some way in which the learning process could be carried out at different computers at the same time, in parallel. However, in this second suggestion, efforts have to be concentrated in order to find the optimum solution.

Finally, this tool could be further improved if a more realistic dynamic model was applied. The control parameters in this project are very rudimentary. In an autopilot that is closer to reality, the module of thrust would be given by the flapping frequency and the stroke swept angle, for instance. Moreover, in a real flapping wing vehicle the direction of thrust does not necessarily coincide with the stroke plane angle, but on the aerodynamic model. Therefore, a deeper knowledge about flapping wing aerodynamics would allow the design of an autopilot that accounts for the control parameters that are characteristic of a real flapping wing MAV.

References

[1] IWM staff. *A brief history of drones*. Imperial War Museum. 2018. URL: https://www.iwm.org.uk/history/a-brief-history-of-drones (visited on 02/15/2018).

[2] EFE. *Así es la nueva normativa de drones en España*. 20 minutos. 2018. URL: https://www.20minutos.es/noticia/3229653/0/drones-nueva-normativa/ (visited on 02/17/2018).

[3] John Excell. *The rise of the micro air vehicle*. The Engineer. 2013. URL: https://www.theengineer.co.uk/issues/aerospace-and-defence-2013/the-rise-of-the-micro-air-vehicle/ (visited on 02/17/2018).

[4] Gryphon Adams. *How do eagles learn to fly?* Mom.me. 2017. URL: http://animals.mom.me/how-do-eagles-learn-to-fly-3179849.html (visited on 02/21/2018).

[5] University of Bristol. *UAV performs first ever perched landing using machine learning algorithms*. Phys.org. 2017. URL: https://phys.org/news/2017-01-uav-perched-machine-algorithms.html (visited on 02/22/2018).

[6] Bernard Etkin & Lloyd Duff Reid. *Dynamics of Flight: Stability and Control*. 3rd edition. John Wiley & Sons, INC, 1996.

[7] Ingrid Hagen Johansen. "Autopilot Design for Unmanned Aerial Vehicles". MA thesis. Trondheim: Norwegian University of Science and Technology, June 2012.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd edition. MIT Press, 2017.

[9] Daniel Morón. *Design of flapping wing kinematics for trimmed flight*. End of Degree Project, Universidad Carlos III de Madrid. June 2017.

[10] Moh Syaifuddin. "Design and evaluation of a LIPCA-actuated flapping device". In: *Smart Materials and Structures* 15 (2006), pp. 1225–1230.

[11] M.D. Mikhailov & A.P. Silva Freire. "The drag coefficient of a sphere: An approximation using Shanks transform". In: *Powder Technology* 237 (2013), pp. 432–435.

[12] Joseph Katz & Allen Plotkin. *Low-Speed Aerodynamics*. 2nd edition. Cambridge University Press, 2001.

Appendix A - Time-step

In order to make sure that the size of a time-step is the proper one for the integration of the dynamics, a brief study has been carried out in the one dimensional problem. Using an inappropriate time-step size would make the simulation be far from the analytic solution, so that information would be missed and results would not be reliable.

A specific trajectory has been plotted with different resolutions for the same period of time (Figure 34). The trajectory has been defined with an initial condition of rest. A sinusoidal evolution of the thrust force has been applied to the vehicle, $T = T_{max}\sin(\frac{1}{2}t)$. In this way, the vehicle state variables show a non-linear evolution, which makes it easy to find an adequate time-step size.

The result of plotting the time evolution at different time-step sizes confirms that $h = 0.05$ is a valid approach since it produces a very smooth graph, very close to $h = 0.025$; there is an error of the order of $O(10^{-4})$ between both curves, which is almost negligible. Therefore, $h = 0.05$ will be set in this problem.

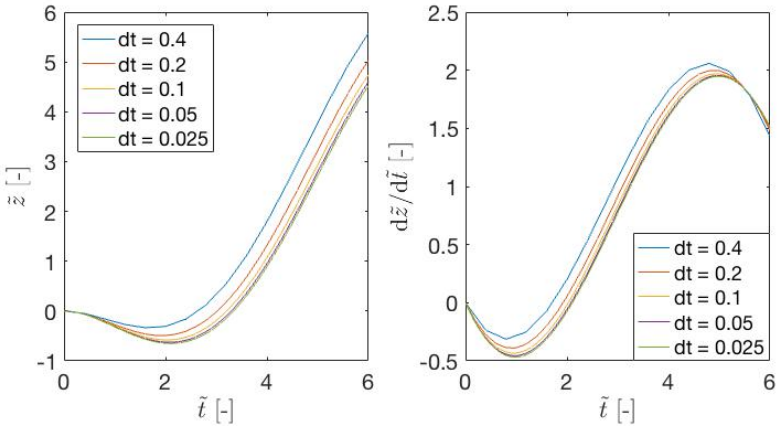


Figure 34: Study of the time-step size in the one dimensional motion case

Appendix B - Project Budget

This appendix shows an estimation about the budget that has been required for the development of this project.

- MATLAB License: This has been the main tool used for the numerical calculations of this project. The price of an individual MATLAB license such as the one that has been used has a price of 2000€ for a company.
- Computational cost: Since the number of iterations needed for the training process in the two and three degrees of freedom problem is very high, they have been carried out with a Work Station. The price of the Work Stattion that has been used in this project is estimated at 1000€.
- Personal computer: The one degree of freedom problem as well as the data processing of the two and three d.o.f. problems have been carried out in a personal computer. The PC used for these tasks has been a *MacBook Pro* with an Intel Core i5, priced at about 1500€.
- Worker salary: A Junior Engineer in Spain is estimated to earn 11€ per hour. Considering that the time spent on this project has been about 500 hours, the total worker salary reaches 5500€.

Finally, the total budget estimated for this project is 10000€.